

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, j, m, v[100];
    cout<<"n=";
    for(i=0; i<=n-1; i++)
        cin>>v[i];
    for (i=0; i<=n-2; i++)
        for (j=i+1; j<=n-1; j++)
            if(v[i]>v[j])
            {
                m=v[i];
                v[i]=v[j];
                v[j]=m;
            }
    for(i=0; i<=n-1; i++)
        cout<<v[i]<<" ";
    return 0;
}
```

Informatică și TIC

Clasa a VIII-a

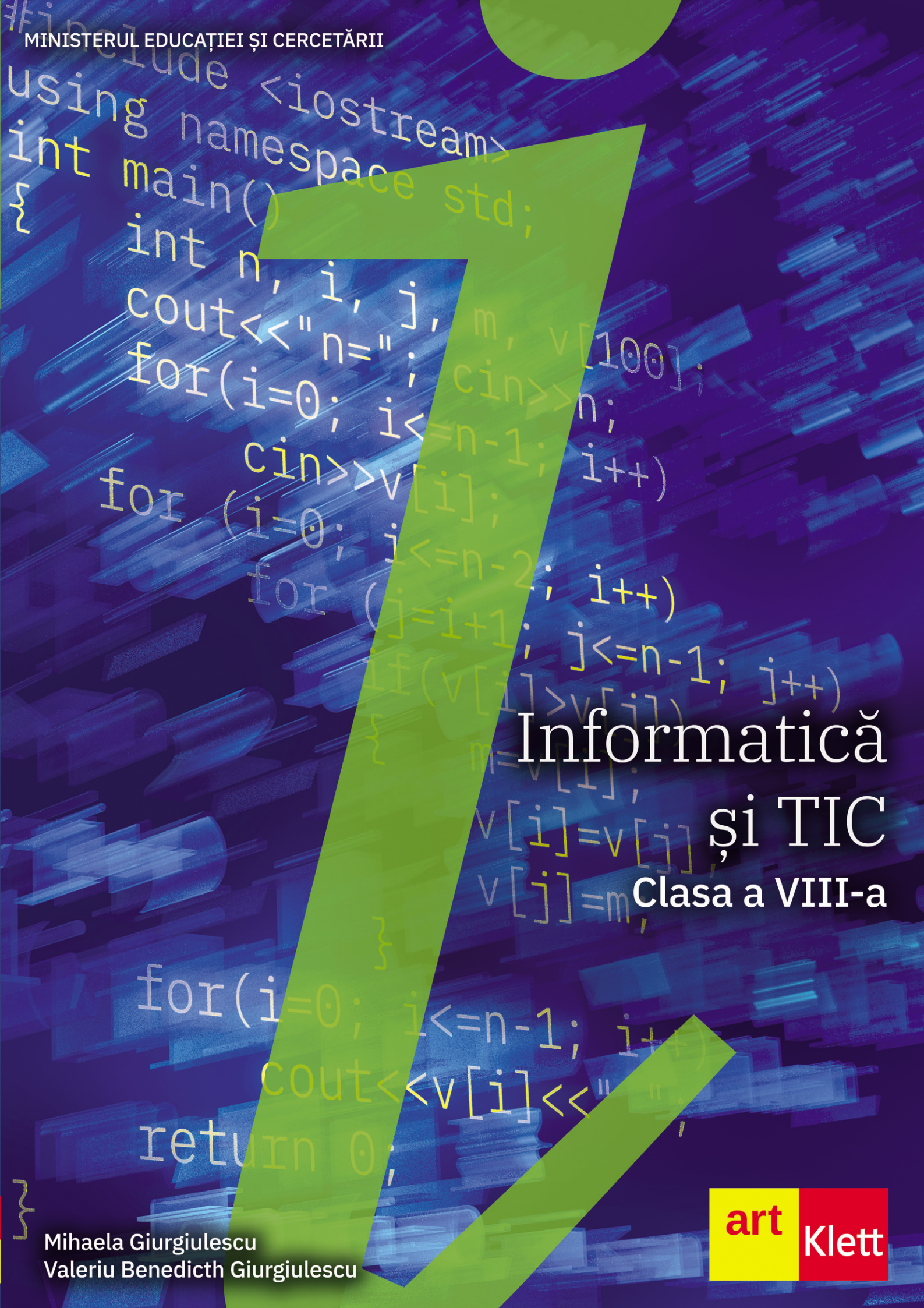
Mihaela Giurgiulescu
Valeriu Benedicth Giurgiulescu

art Klett

Acest manual este proprietatea Ministerului Educației și Cercetării.

Acest proiect de manual școlar este realizat în conformitate cu *Programa școlară aprobată prin Ordinul ministrului educației naționale nr. 3393/28.02.2017.*

116.111 – numărul de telefon de asistență pentru copii



```
#include <iostream>
using namespace std;
int main()
{
    int n, i, j, m, v[100];
    cout<<"n=";
    for(i=0; i<=n-1; i++)
        cin>>v[i];
    for (i=0; i<=n-2; i++)
        for (j=i+1; j<=n-1; j++)
            if(v[i]>v[j])
            {
                m=v[i];
                v[i]=v[j];
                v[j]=m;
            }
    for(i=0; i<=n-1; i++)
        cout<<v[i]<<" ";
    return 0;
}
```

Informatică și TIC

Clasa a VIII-a

Mihaela Giurgiulescu
Valeriu Benedicth Giurgiulescu

art Klett

Manualul școlar a fost aprobat de Ministerul Educației și Cercetării prin ordinul de ministru nr. 5615/23.09.2020.

Manualul este distribuit elevilor în mod gratuit, atât în format tipărit, cât și în format digital, și este transmisibil timp de patru ani școlari, începând din anul școlar 2020 – 2021.

Inspectoratul Școlar

Școala/Colegiul/Liceul

ACEST MANUAL A FOST FOLOSIT DE:

Anul	Numele elevului	Clasa	Anul școlar	Aspectul manualului*			
				format tipărit		format digital	
				la primire	la predare	la primire	la predare
1							
2							
3							
4							

* Pentru precizarea aspectului manualului se va folosi unul dintre următorii termeni: *nou*, *bun*, *îngrijit*, *neîngrijit*, *deteriorat*.

- Cadrele didactice vor verifica dacă informațiile înscrise în tabelul de mai sus sunt corecte.
- Elevii nu vor face niciun fel de însemnări pe manual.

Referenți științifici:

lector univ. dr. Alexandra Băicoianu, Facultatea de Matematică și Informatică, Universitatea „Transilvania” din Brașov
profesor gradul I Victor Manz, Colegiul Național de Informatică „Tudor Vianu”, București

Redactor-șef: Roxana Jeler

Corector: Theodor Zamfir

Design: Faber Studio

Tehnoredactor: Monica Manolache

Copertă: Alexandru Daș, Anca Chiriță

Activități digitale interactive și platformă e-learning: Infomedia Pro: www.infomediapro.ro

Înregistrări și procesare sunet: Infomedia Pro

Voce: Andreea Gaică, Mihaela Giurgulescu

Credite foto și video: Dreamstime, Wikimedia Commons, Pixabay, Wikipedia, Freepik

Animații: Infomedia Pro

ISBN 978-606-076-472-4

Pentru comenzi vă puteți adresa Departamentului Difuzare

C.P. 12, O.P. 63, cod 014880, sector 1, București

Telefoane: 021.796.73.83, 021.796.73.80

Fax: 021.369.31.99

www.art-educational.ro

Toate drepturile asupra acestei lucrări sunt rezervate Editurii Art Klett.

Nicio parte a acestei lucrări nu poate fi reprodusă, stocată ori transmisă, sub nicio formă (electronic, mecanic, fotocopiare, înregistrare sau altfel), fără acordul prealabil scris al Editurii Art Klett.

© Editura Art Klett, 2022

Cuvânt-înainte

Manualul este conceput din dorința de a vă ajuta să folosiți cât mai bine tehnologia, pentru a vă bucura din plin de beneficiile acesteia, și cu speranța că va reprezenta pentru voi un material de calitate, frumos și ușor de utilizat.

Lucrarea este structurată în trei părți: prima parte conține teme legate de utilizarea programelor de calcul tabelar, cu detalierea programului MS Excel, a doua parte este dedicată realizării de site-uri web, cu evidențierea modalității de lucru colaborativ, iar ultima parte este dedicată realizării de algoritmi și programe informatice pentru lucrul cu șirurile de valori numerice, utilizând limbajul de programare C++ și mediul de dezvoltare integrat Code::Blocks. Nu lipsește nici o exemplificare de utilizare a șirurilor în mediul Open Roberta Lab, utilizând un robot didactic virtual, Calliope Mini.

Lecțiile conțin aplicații practice care favorizează învățarea prin descoperire. De asemenea, exercițiile propuse la fiecare lecție conțin exemple de rezolvare, pentru o înțelegere mai ușoară a enunțului și se rezolvă pe baza informațiilor transmise în manual. Pe parcursul lucrării, la finalul fiecărei părți sunt prevăzute activități de recapitulare, pentru a consolida cunoștințele acumulate, și de autoevaluare, pentru ca fiecare dintre voi să poată să-și aprecieze gradul de înțelegere a noțiunilor. Există și o recapitulare finală care conține itemi corespunzători noțiunilor învățate pe parcursul întregului an școlar. Toate exercițiile cuprinse în autoevaluări și în recapitularea finală sunt rezolvate la sfârșitul manualului.

Și pentru că sunteți într-un an important pentru voi, anul tranziției spre liceu, vă îndemnăm să țintiți cât mai sus în tot ceea ce faceți, pentru că „nu vei zări niciodată un curcubeu dacă te uiți întotdeauna în jos“ (Charlie Chaplin).

Autorii

Manualul cuprinde:

Varianta tipărită



Varianta digitală (similară cu cea tipărită, care cuprinde, în plus, activități multimedia interactive de învățare, cu rolul de a spori valoarea cognitivă). Varianta digitală este accesibilă pe platforma www.manuale.edu.ro.

Cum sunt marcate activitățile multimedia interactive de învățare (AMII)?

- Activitate statică, de ascultare activă și observare dirijată a unei imagini semnificative
- Activitate animată (filmuleț sau scurtă animație)
- Activitate interactivă, de tip exercițiu sau joc, în urma căreia elevul are feedback imediat

Ce propune acest manual

Manualul propune ghidarea elevilor, asemenea unui tutorial interactiv și prietenos, în lumea fascinantă a tehnologiei informației și comunicațiilor.

El este structurat în trei mari unități de învățare și răspunde viziunii și cerințelor noului curriculum. Prima unitate acoperă integral, în cele 6 lecții existente, domeniul de conținut din programa școlară *Calcul tabelar*. Urmează a

doua unitate, structurată în 4 lecții, ce prezintă în detaliu conținutul despre *Pagini web*, în timp ce ultima unitate dezvoltă, în 12 lecții, conținutul despre *Algoritmi*.

Nu puteau lipsi recapitulările sumative și autoevaluările, marcate distinct pentru fiecare unitate, precum și o recapitulare finală, urmată de răspunsuri la multe dintre problemele și exercițiile propuse. Toate unitățile manualului



Structura unităților de învățare

Lecții de predare-învățare

Modul de abordare modern facilitează învățarea și asigură, totodată, stimularea creativității elevilor care vor folosi cunoștințele dobândite în realizarea unor mini proiecte, atât individuale, cât și colaborative, pe teme de interes caracteristice vârstei lor (respectând creditarea informației și drepturile de autor). Principalele rubrici ale lecțiilor sunt marcate diferit și consecvent pe parcursul manualului.

Descoperiți prezintă, într-o manieră plăcută și graduală, introducerea conținutului nou de învățat. Sunt oferite numeroase exemple, iar conținuturile noi sunt prezentate cu suportul multor imagini reprezentative ce sunt explicate în detaliu, unde este cazul.

Aplicați și investigați asigură aplicarea noțiunilor noi, elevii fiind ghidați pas cu pas în realizarea unor sarcini specifice. Sunt implicați, astfel, în mod activ în procesul de învățare, cunoștințele căpătând un caracter operațional accentuat.

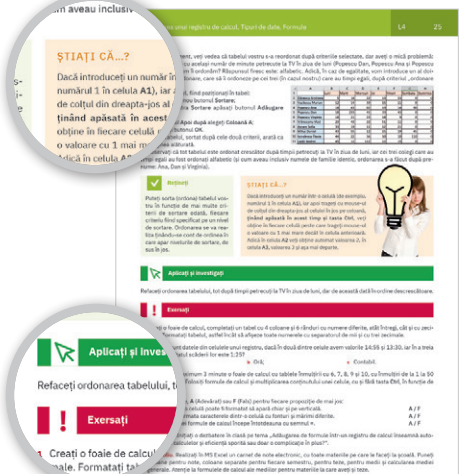
Exersați propune itemi ce vor consolida învățarea, din care nu lipsesc temele de portofoliu, dezbaterile în clasă, lucrul în echipe sau individual, activitățile de documentare/investigare și realizarea unor eseuri scurte sau a unor sarcini interesante.

Amintiți-vă și Rețineți sunt alte rubrici care apar pe parcursul lecțiilor, atunci când conținutul prezentat necesită reamintirea unor noțiuni sau când se evidențiază conținutul important, de bază pentru achizițiile viitoare.

Știați că... prezintă unele informații sau curiozități despre aplicațiile descrise în manual, astfel cei interesați să afle mai multe o pot face apelând la cuvintele-cheie din lecții.

Portofoliu/Dezbateri

Pe parcursul lecțiilor sunt recomandate mai multe teme de **dezbateri** și pentru **portofoliul elevilor**. Metode moderne de învățare-evaluare, acestea sunt prezentate în cadrul rubricii **Exersați**. Proiectele sunt fie individuale, fie în echipă, având atât caracter informativ, cât și formativ. Se propun teme pentru documentări și investigații, manualul oferind, de asemenea, o serie generoasă de exemple de cum putem folosi diverse aplicații online sau instalate pe propriile calculatoare.



Manualul este structurat în trei unități

oferă numeroase activități interdisciplinare însoțite de indicații și modele de rezolvare.

Manualul de față insistă asupra utilizării responsabile și, totodată, eficiente a TIC. Specificul disciplinei implică utilizarea unor metode didactice activ-participative care sunt bogat sugerate de activitățile prezentate în acest manual, astfel încât să fie atinse toate cele trei competențe generale vizate.

Recapitulare

La fiecare final de capitol, înainte de testul de autoevaluare aferent, se află câte un test de recapitulare. Probele concepute conțin selecții de itemi obiectivi, semiobiectivi și subiectivi, cu nivel de la simplu la complex. Recapitularea vizează atât corpusul de noțiuni învățate, cât și capacitățile asociative, de interpretare și aplicare a cunoștințelor în contexte noi.

Autoevaluare

Autoevaluările sunt însoțite de punctajele de notare și de răspunsuri care se află la finalul manualului, elevii putând astfel să se verifice singuri.

Manualul propune și o recapitulare finală, din toată materia parcursă de-a lungul anului școlar. Exercițiile și problemele date sunt astfel concepute încât să asigure o bună verificare a cunoștințelor acumulate, precum și operaționalizarea acestora în situații noi, variate. Și aici, elevii au la dispoziție, la sfârșitul manualului, răspunsurile la exercițiile propuse, pentru a se putea autoevalua cu ușurință, obținând un feedback imediat.



	Pag.	Lecții
	8	Să ne amintim! Scurtă recapitulare a cunoștințelor de până acum
UNITATEA 1		
Calcul tabelar	12	L1: Utilizarea foilor de calcul tabelar
	14	L2: MS Excel – elemente de interfață
	18	L3: Operații principale cu un registru de calcul
	21	L4: Formatarea unui registru de calcul. Tipuri de date. Formule
	26	L5: Modalități de utilizare a registrelor de calcul. Calcule tabelare, funcții
	29	L6: Modalități grafice de utilizare a registrelor de calcul
	32	Recapitulare
	33	Autoevaluare
UNITATEA 2		
Aplicații de realizare a paginilor web	36	L1: Pagini web – exemple, analiză, structură
	40	L2: Aplicații specializate pentru crearea paginilor web – elemente de interfață și facilități
	45	L3: Crearea unei pagini web. Publicarea unui site de o pagină
	51	L4: Construirea unui site Internet cu mai multe pagini. Lucrul colaborativ
	54	Recapitulare
	55	Autoevaluare
UNITATEA 3		
Algoritmi și programare	58	L1: Structura decizională (alternativă) – recapitulare
	62	L2: Structura repetitivă – recapitulare
	66	L3: Șiruri de valori (noțiuni introductive)
	68	L4: Tablou unidimensional. Operații cu șiruri de valori: citire, afișare, parcurgere
	72	L5: Verificarea unor proprietăți ale elementelor unui șir de valori
	76	L6: Algoritmi de numărare
	80	L7: Operații matematice bazate pe valoarea elementelor unui șir
	84	L8: Căutarea unui element într-un șir *
	87	L9: Sortarea (ordonarea) șirurilor *
	90	L10: Inserarea/ștergerea elementelor unui șir de valori *
	94	L11: Probleme cu două sau mai multe șiruri
	100	L12: PROIECT – Utilizarea șirurilor de numere în robotică
	102	Recapitulare
	103	Autoevaluare
104	Recapitulare finală	
107	Răspunsuri (selectiv)	

Competențe specifice vizate

1.1., 3.1.

Competențe generale

1. Utilizarea responsabilă și eficientă a tehnologiei informației și comunicațiilor
2. Rezolvarea unor probleme elementare prin metode intuitive de prelucrare a informației
3. Elaborarea creativă de mini proiecte care vizează aspecte sociale, culturale și personale, respectând creditarea informației și drepturile de autor

Competențe specifice

- 1.1. Utilizarea foilor de calcul tabelar în vederea rezolvării unor situații problemă simple
- 1.2. Utilizarea unui editor dedicat pentru realizarea unor pagini web cu diverse teme
- 2.1. Identificarea șirurilor de valori în diferite contexte de prelucrare în vederea construirii algoritmilor
- 2.2. Rezolvarea unor probleme simple prin construirea unor algoritmi de prelucrare a șirurilor de valori
- 3.1. Elaborarea de produse informatice utilizând aplicații de calcul tabelar
- 3.2. Elaborarea/actualizarea de pagini web conform unor specificații date
- 3.3. Implementarea algoritmilor într-un mediu de programare

1.2., 3.2.

2.1., 2.2., 3.3.

Notă: *Pe parcursul manualului, conținuturile marcate prin asterisc (*) aprofundează/extind programa școlară.*



Să ne amintim! Scurtă recapitulare a cunoștințelor de până acum



Amintiți-vă

Principalele operații în lucrul cu fișierele și folderurile sunt: creare, deschidere, editare, salvare, închidere, selectare, copiere, mutare, ștergere.

Pentru lucrul cu elementele conținute în fișiere, principalele operații sunt: selectare, copiere, mutare, ștergere.

Comenzile corespunzătoare operațiilor de mai sus se găsesc, în general, la majoritatea programelor, sub formă de opțiuni de meniu, în meniurile *File (Fișier)* și *Edit (Editare)*.

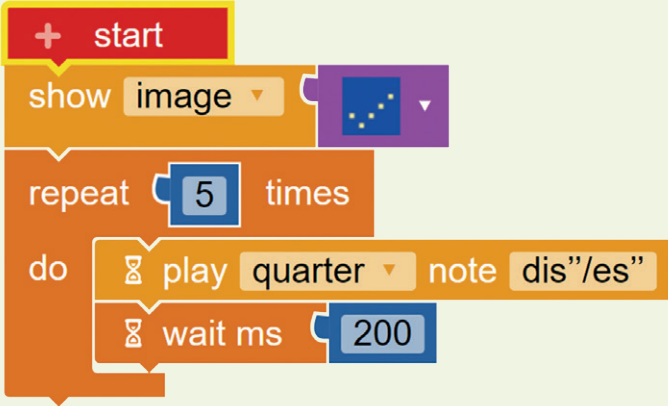
Pentru lucrul mai eficient, puteți folosi comenzile rapide (combinațiile de taste) următoare (valabile atât în editoarele de text, de exemplu, în MS Word, cât și în programele de editare de sunete, în programele de editare video și în editorul de C++):

- **Creare (New)** – Ctrl+N;
- **Deschidere (Open)** – Ctrl+O;
- **Salvare (Save)** – Ctrl+S;
- **Închidere (Close sau Exit)** – Alt+F4;
- **Selectare integrală (Select All)** – Ctrl+A;
- **Copiere (Copy)** – Ctrl+C;
- **Mutare** – aceasta este o combinație de două acțiuni:
Decupare din locația inițială (Cut) – Ctrl+X și **Lipire în noua locație (Paste)** – Ctrl+V;
- **Ștergere (Delete)** – tasta DEL sau Delete (în funcție de producător, pe tasta respectivă marcajul poate să difere);
- **Anulează (Undo)** – Ctrl+Z (pentru a reveni la starea dinaintea ultimei acțiuni efectuate).



Exersați

Sisteme de operare	Dați trei exemple de sisteme de operare foarte utilizate în prezent.
Realizarea unui document electronic	Realizați în MS WORD o scurtă prezentare electronică, pe tema „Cinci lucruri importante pe care le-am învățat la Informatică și TIC în clasa a VII-a”. Folosiți în prezentare texte și imagini. Titlul să fie centrat și îngroșat.
Siguranța pe Internet	Precizați trei principii de bază care vă asigură o navigare pe Internet lipsită de pericole.
Poșta electronică	Care sunt cele două categorii de programe prin intermediul cărora puteți utiliza poșta electronică?
Reguli de comunicare online	<i>Neticheta</i> impune norme de comportament general în mediul online. Enumerați cinci reguli specifice comunicării în acest mediu.
Operații cu sunete	Care sunt operațiunile de bază pentru a obține un material audio de bună calitate (de exemplu, un interviu de două minute luat unui coleg)?
Aplicații video	Numiți două aplicații dedicate prelucrărilor video. Realizați un videoclip cu durata între 30 s și 3 min, pe o temă ecologică sau educațională (de exemplu, „Parcurile din jurul nostru îngrijite de copiii de la școlile din localitate”), după următorul scenariu: <ul style="list-style-type: none"> • să fie o succesiune de secvențe filmate și/sau fotografii pe tema aleasă (se pot folosi inclusiv fotografii ale panourilor expuse în școli, panouri realizate de copii pentru proiectele la care au participat);

	<ul style="list-style-type: none"> • dacă se folosesc fotografiile, acestea să fie în număr de cel puțin zece; • dacă sunt folosite fotografiile sau secvențe filmate luate de pe Internet, se vor folosi numai din domeniul public, liber de drepturi, sau cu acordul scris al autorilor; • să se pună mare, pe ecranul de început, titlul videoclipului; • să se pună efecte de tranziții între cadre; • să se adauge un fundal muzical adecvat care să se întindă pe durata întregului videoclip. <p>Videoclipul să fie exportat la final în format .mp4.</p>
Aplicații colaborative	<p>Lucru în echipă: Realizați, la nivelul clasei, o carte electronică cu titlul <i>Povești, povestiri, amintiri</i>. Parcurgând toate acțiunile următoare, veți obține un minunat proiect realizat împreună:</p> <ul style="list-style-type: none"> • Mai întâi, alegeți un „manager de proiect“. El va iniția fișierul Word în care cu toții veți introduce câte ceva pentru realizarea cărții. • Salvați fișierul pe Google Drive, cu titlul „Povești, povestiri, amintiri.doc“. • Faceți ceea ce este necesar pentru ca toți elevii clasei să aibă acces la fișierul care va conține cartea: <ul style="list-style-type: none"> – Asigurați-vă că aveți câte un cont Google pentru fiecare. – Managerul de proiect va da tuturor drept de acces la acel fișier. • Scrieți, fiecare dintre voi, în documentul creat, o poveste, o povestire, o amintire care vi se pare interesantă sau hazlie și merită împărtășită. Puteți adăuga imagini și formata textul așa cum doriți.
GIS	<p>Ce este și la ce se folosește un sistem GIS? Dați un exemplu de utilizare practică în viața de zi cu zi.</p> <p>Plimbați-vă prin zona din jurul casei (pe o rază de 1 km), pe jos sau cu bicicleta, și adăugați pe Google Maps toate cuiburile de barză pe care le găsiți sau orice alte obiective doriți. Așteptați să fie avizate și să devină disponibile toate elementele adăugate, apoi formați grupuri de lucru și analizați ce elemente a adăugat fiecare, precizia localizării și multitudinea amănuntelor din descriere, precum și eleganța descrierii.</p>
Programarea roboților	<p>Explicați ce face Calliope mini, în mediul de programare Open Roberta Lab, dacă primește următorul program:</p>  <pre> + start show image [European Union flag] repeat 5 times do play quarter note "dis"/es" wait ms 200 </pre>
Algoritmi și programare	<p>Realizați în C++ programul care afișează dublul celei mai mici cifre a unui număr întreg introdus de la tastatură. <i>Indicație: Intrare: 345, Ieșire: 6; Intrare: -57, Ieșire: 10.</i></p> <p>Faceți un program în C++ care calculează și afișează aria și perimetrul unui triunghi echilateral cu latura n, unde n număr natural mai mic decât 200.</p>

U1 Calcul tabelar

L1 Utilizarea foilor de calcul tabelar

L2 MS Excel – elemente de interfață

L3 Operații principale cu un registru de calcul

L4 Formatarea unui registru de calcul. Tipuri de date. Formule

L5 Modalități de utilizare a registrelor de calcul. Calcule tabelare, funcții

L6 Modalități grafice de utilizare a registrelor de calcul

Recapitulare

Autoevaluare



tablet news 1.02.2015

Mauris rutrum, tellus vitae sodales vulputate, nunc lacus placerat ipsum, ut fringilla nulla felis vel dolor. Sed sem tortor, dapibus nec euismod at, semper semper sem.



Mauris rutrum, tellus vitae sodales vulputate, nunc lacus placerat ipsum, ut fringilla nulla felis vel dolor. Sed sem tortor, dapibus nec euismod at, semper semper sem.



Mauris rutrum, tellus vitae sodales vulputate, nunc lacus placerat ipsum, ut fringilla nulla felis vel dolor. Sed sem tortor, dapibus nec euismod at, semper semper sem.

Utilizarea foilor de calcul tabelar



Descoperiți

▶ Foile de calcul electronice reprezintă o modalitate de gestionare a datelor organizate sub forma unor tabele digitale. Numele de „foaie de calcul“ provine din contabilitate, domeniu în care se folosesc formulare de hârtie de mari dimensiuni, liniate orizontal și vertical sub formă de tabel.

Utilizarea calculatorului a adus mai multe avantaje în calculul tabelar: a permis construirea ușoară a tabelului (inserare/ștergere de rânduri și coloane), adăugarea la căsuțele din tabele, pe lângă valorile înscrise în ele, a formulelor de calcul pentru aceste valori, a specificațiilor de afișare a valorilor etc.

Exemplu practic: faceți un tabel cu materiile voastre de la școală și cu notele dintr-un semestru. Pentru exemplul de mai jos vom considera 15 materii, fără teze, pentru un semestru, câte 5 note la fiecare materie. Vom calcula media pentru fiecare materie (într-o coloană separată) și media generală la final.

Dacă am dori să vedem ce impact ar avea asupra mediei generale o notă mai mare la Geografie (de exemplu, 9 în loc de 7), ar trebui să recalculăm media la Geografie, apoi să recalculăm media generală.

Imaginați-vă acum că media pentru fiecare materie este calculată automat de calculator, în baza unei formule pe care am introdus-o acolo unde trebuie să apară media, iar pentru media generală se procedează la fel (se calculează automat în baza unei formule). Dacă am schimba nota 7 din tabelul nostru cu nota 9, s-ar modifica imediat și automat atât media la Geografie, cât și media generală. Iar noi am vedea rapid ce diferență ar aduce la media generală o notă de 9 în locul notei de 7, fără să mai facem calcule. Această posibilitate este creată, alături de multe altele, de apariția foilor de calcul electronice și a programelor dedicate pentru lucrul cu ele.

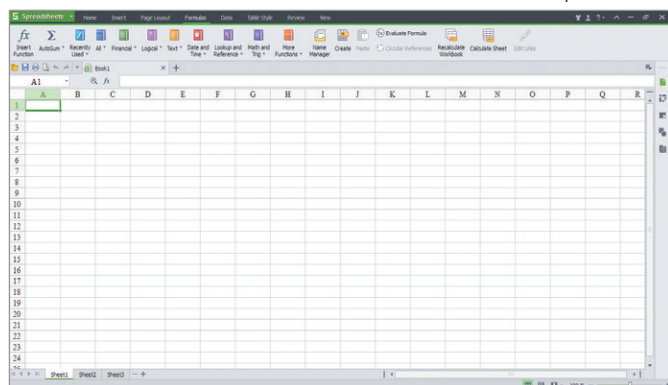
	A	B	C	D	E	F	G
1	MATERIA	NOTA 1	NOTA 2	NOTA 3	NOTA 4	NOTA 5	MEDIA
2	Matematica	10	10	10	9	10	9.80
3	Romana	10	10	10	10	10	10.00
4	Istorie	10	10	10	10	10	10.00
5	Fizica	9	9	9	9	9	9.00
6	Desen	10	10	10	10	10	10.00
7	Muzica	10	10	10	10	10	10.00
8	Sport	10	10	10	10	10	10.00
9	Geografie	10	10	7	10	10	9.40
10	Religie	10	10	10	10	10	10.00
11	Limba engleză	10	10	10	10	10	10.00
12	Limba Germană	10	10	10	10	10	10.00
13	Chimie	10	10	10	10	10	10.00
14	TIC	10	10	10	10	10	10.00
15	Optional informatica	10	10	10	10	10	10.00
16	Biologie	9	9	9	9	9	9.00
17							
18						MEDIA GENERALA:	9.81

Ce observați la tabelul de mai sus? Pe două dintre margini regăsim niște marcaje ca la șah: pe o latură numere și pe o latură perpendiculară, litere. Acestea ne folosesc atunci când vrem să ne referim sau să ne poziționăm pe o anumită căsuță din tabel. De exemplu, căsuța în care este scris „TIC“ este A14 (intersecția dintre coloana A și rândul 14 din tabel), iar cea în care e calculată media generală este G18.

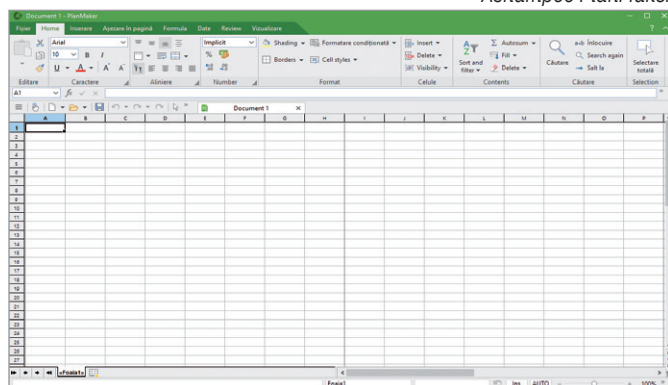
Există o multitudine de programe dedicate lucrului cu foi de calcul (cunoscute și sub numele de programe de calcul tabelar). Amintim aici doar câteva dintre cele mai populare:

- MS Excel (din pachetul de programe Microsoft Office) sau Excel Online
- LibreOffice Calc (din pachetul de programe LibreOffice)
- OpenOffice Calc (din pachetul de programe Apache OpenOffice)
- WPS Spreadsheets (din pachetul de programe WPS Office)
- Ashampoo PlanMaker (din pachetul de programe Ashampoo Office)
- Google Sheets (Foi de calcul Google)

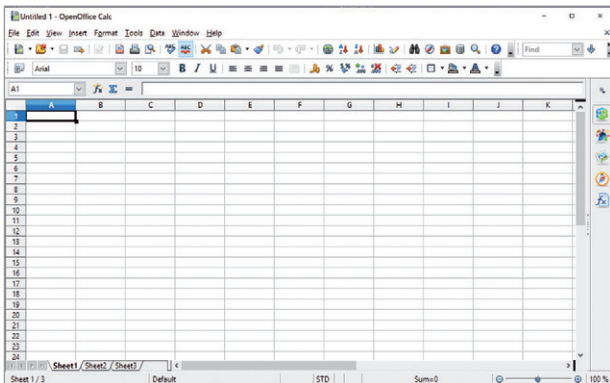
WPS Spreadsheets



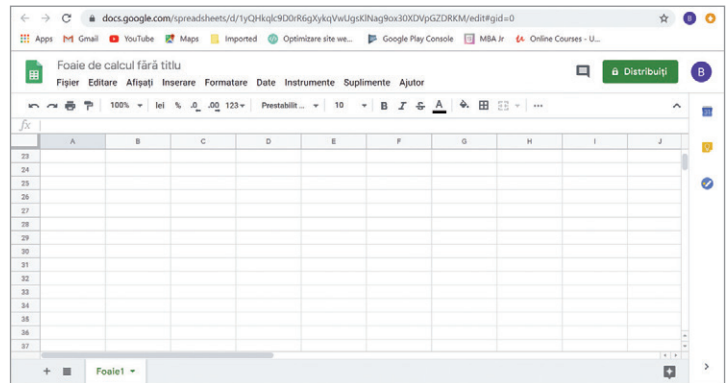
Ashampoo PlanMaker



OpenOffice Calc



Google Sheets



Fiecare dintre programele specializate în lucrul cu foi de calcul are o serie de particularități care pot fi avantaje sau dezavantaje, în funcție de fiecare dintre noi: **MS Excel** face parte din cunoscutul pachet de programe MS Office și este, poate, cel mai popular, astfel că putem găsi cu ușurință ajutor la cei din jurul nostru, fiind disponibil și în limba română; **Calc** (din pachetul LibreOffice) și **Spreadsheets** (din pachetul WPS Office) sunt aplicații gratuite, la îndemâna oricui și sunt disponibile și în limba română; unele se instalează pe calculator și le aveți la îndemână oricând, indiferent dacă aveți legătură la Internet sau nu, altele sunt disponibile online, nu necesită instalare și nu ocupă spațiu de stocare pe calculatoarele voastre, dar necesită legătură la Internet, altfel nu pot fi accesate.

ȘTIAȚI CĂ...?

În aprilie 1975, Bill Gates împreună cu Paul Allen au înființat compania Microsoft care a dezvoltat atât cunoscutul sistem de operare Microsoft Windows, cât și pachetul de programe Microsoft Office, din care fac parte Word, PowerPoint, dar și Excel.



Aplicații și investigații

Căutați pe Internet informații despre cele șapte programe menționate anterior. Faceți în caiete un tabel în care să aveți câte o coloană pentru fiecare program și câte o linie în care să analizați aspectele următoare:

- 1 din ce pachet de programe face parte;
- 2 este gratuit sau contra cost (dacă utilizarea presupune costuri, treceți prețul în tabel);
- 3 pentru ce sisteme de operare este disponibil;
- 4 este disponibil sau nu și în limba română;
- 5 se instalează sau e disponibil online;
- 6 acceptă lucrul colaborativ (pot lucra mai multe persoane online pe același fișier);
- 7 ați auzit de el de la prieteni, colegi sau părinți;
- 8 ați mai folosit alte programe din același pachet de programe;
- 9 există versiunea programului și pentru mobile.

Priviți comparativ caracteristicile programelor. Care dintre ele credeți că sunt mai importante? Ce program vi s-ar părea mai potrivit pentru voi?



Exersați

- 1 Dați exemple de trei programe pentru calcul tabelar.
- 2 Argumentați de ce este ușor să facem modificări într-o foaie de calcul realizată cu ajutorul unui program specializat.
- 3 Scrieți, în caiete, **A** (Adevărat) sau **F** (Fals), în funcție de valoarea de adevăr a propozițiilor:
 - a Un program de calcul tabelar produce fișiere video. A / F
 - b Un tabel realizat cu ajutorul unui program de calcul tabelar poate fi ușor modificat și tipărit de nenumărate ori. A / F
 - c Într-un program de calcul tabelar se pot introduce și formule pentru a efectua automat anumite calcule. A / F
- 4 Care sunt avantajele utilizării foilor de calcul pe calculator? Argumentați.
- 5 **Portofoliu.** Faceți un tabel în caiete cu materiile din clasa a VIII-a, cu cele două medii semestriale de la fiecare materie, cu media anuală la fiecare materie și cu media generală. Cronometrați-vă și notați timpul necesar.

MS Excel – elemente de interfață



Descoperiți

Dintre aplicațiile dedicate lucrului cu foile de calcul vom folosi **Microsoft Office Excel** (prescurtat **MS Excel** sau, simplu, **Excel**), componentă a pachetului de programe Microsoft Office. O foaie de calcul care este lucrată în această aplicație se salvează într-un fișier având una dintre următoarele extensii: **.xls**, **.xlsx**, **.xlsm**. În continuare, pentru exemplificări, va fi utilizată versiunea Microsoft Office Professional Plus 2019, în limba română.



Mai departe, vom descoperi împreună cele mai importante elemente ale interfeței aplicației MS Excel.

Aplicația MS Excel se deschide cu un ecran de start. De aici putem alege:

- crearea unui document nou, necompletat: se alege **Nou, Registrul de lucru necompletat**;
- deschiderea unui registru de lucru realizat anterior: se selectează un fișier din lista propusă (cu cele mai recent accesate) sau se caută, în cloud (în contul personal OneDrive) sau pe PC, fișierul dorit;
- parcurgerea unor tutoriale generale despre MS Excel (vă recomandăm să le parcurgeți și voi).

Structura unui registru de calcul (foaie de calcul, coloană, rând, celulă, adresă de celulă)

Aplicația Microsoft Excel organizează datele într-o **foaie de calcul** (*worksheet*) care reprezintă un singur tabel cu mai multe rânduri (linii) etichetate cu numere (1, 2, ...) și coloane etichetate cu litere (A, B, ...). Numărul de linii și de coloane ale unei foi de calcul se mărește dinamic, de la valorile implicite, până la numărul maxim de 1 048 576 de rânduri și 16 384 de coloane.

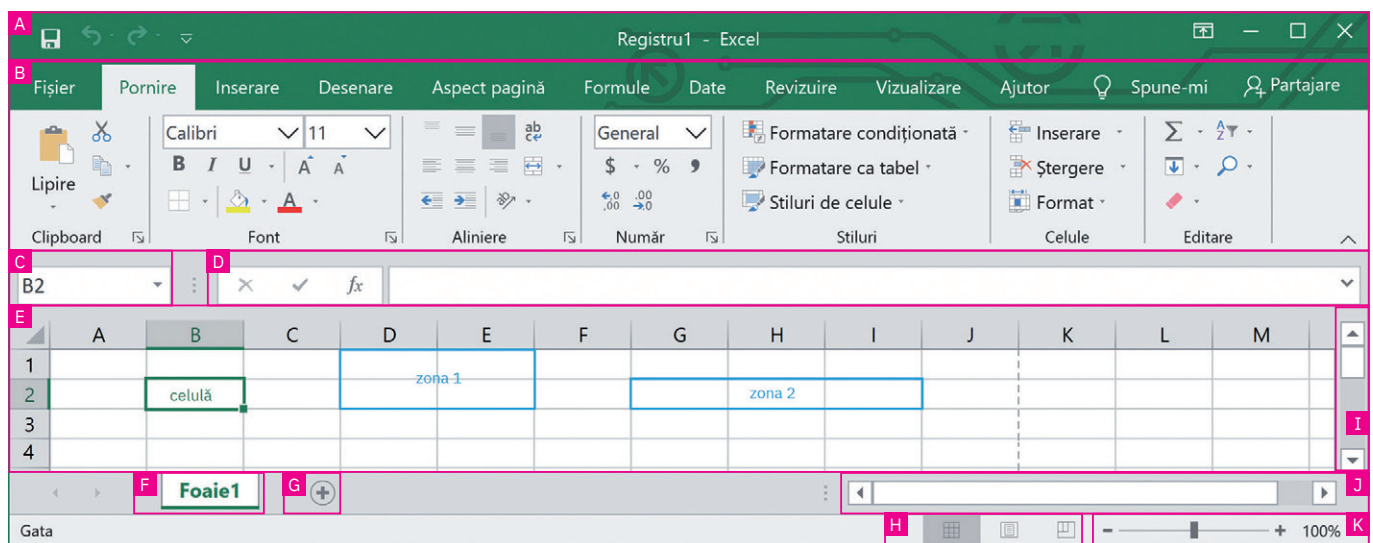
Unitatea elementară adresabilă din tabel este **celula**, aflată la intersecția unui rând și a unei coloane din tabel. În celule se introduc de la tastatură date și formule de calcul pentru obținerea de rezultate. O celulă este accesată prin eticheta coloanei și a rândului la a căror intersecție se află. De exemplu, celula **B2** este celula de la intersecția coloanei **B** și a rândului 2 din tabel (a se vedea imaginea de mai jos).

Mai multe celule adiacente, cu coloane sau rânduri alăturate, formează o **zonă**. Celulele din acea zonă se prelucrează simultan, în același mod. O zonă se adresează prin coordonatele celulelor din colțurile sale stânga-sus și dreapta-jos, despărțite prin „:”. De exemplu, în imaginea de mai jos, zona 1 are coordonatele **D1:E2**, zona 2 **G2:I2**.

Mai multe foi de calcul formează împreună un **registru de calcul**.

Detalierea elementelor de interfață ale aplicației MS Excel

Vom alege să creăm un fișier nou, necompletat. Alegerea făcută inițiază deschiderea în interfața MS Excel a unui fișier care este denumit automat „**Registrul1**”. El este acum în lucru, nesalvat încă.



Fereastra interfeței utilizator a Microsoft Excel este alcătuită din următoarele elemente principale:

A – Bara de titlu

B – Panglica

C – Caseta de nume – afișează numele celulei în care suntem poziționați (în cazul de mai sus, celula este **B2**, marcată cu chenar verde).

D – Bara de formule

E – Zona de lucru (foaia de calcul)

F – Eticheta foi de calcul – aici vom regăsi toate etichetele foilor de calcul create în registrul aflat în lucru – în exemplul de mai sus avem o singură foaie de calcul, numită (implicit) „Foaie1“.

G – Buton de adăugare a unei noi foi de calcul

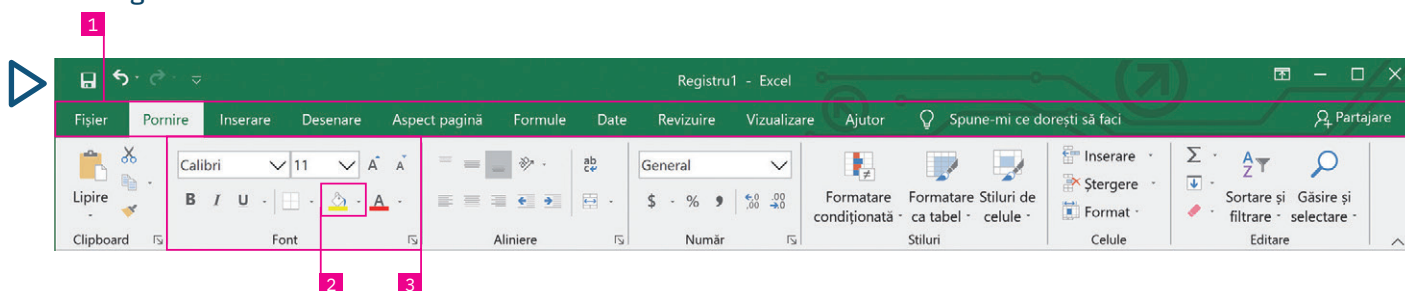
H – Zona de control al modului de vizualizare a tabelului – oferă 3 opțiuni: Normal (implicit), Aspect pagină, Examinare sfârșit de pagină

I – Bara de defilare verticală

J – Bara de defilare orizontală

K – Instrumentul de panoramare (Zoom) – permite mărirea/micșorarea zonei vizibile a tabelului. Pune la dispoziție două butoane („+“ pentru mărire și „-“ pentru micșorare), un potențiomtru care se poate muta cu ajutorul mouse-ului (la stânga pentru micșorare și la dreapta pentru mărire) și o zonă de afișare a nivelului de panoramare (zoom), în partea dreaptă a zonei (în procente).

Panglica – detalieri



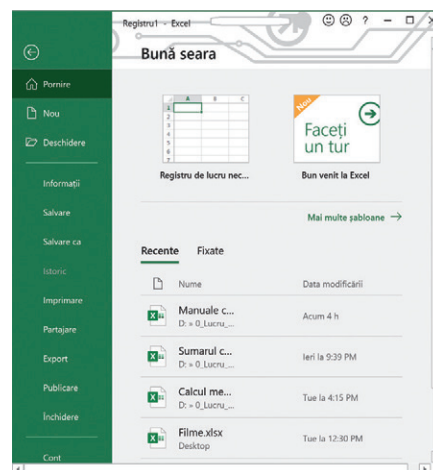
Panglica – include toate comenzile aplicației MS Excel grupate în file, pe tipuri de acțiuni.

Așadar, în panglică regăsim mai multe **file** (1) (Fișier, Pornire, Inserare etc.). Fiecare filă are mai multe **butoane pentru comenzi** (2), grupate în **grupuri de comenzi** (3) cu nume sugestive (imaginea de mai sus).

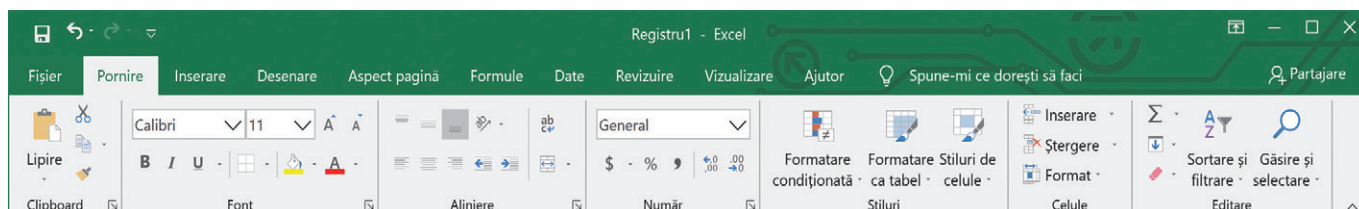
Dacă stați cu mouse-ul deasupra unui buton din interfața Excel, se afișează un text de ajutor cu descrierea funcționalității butonului respectiv.

Principalele file prezente pe panglica MS Excel sunt:

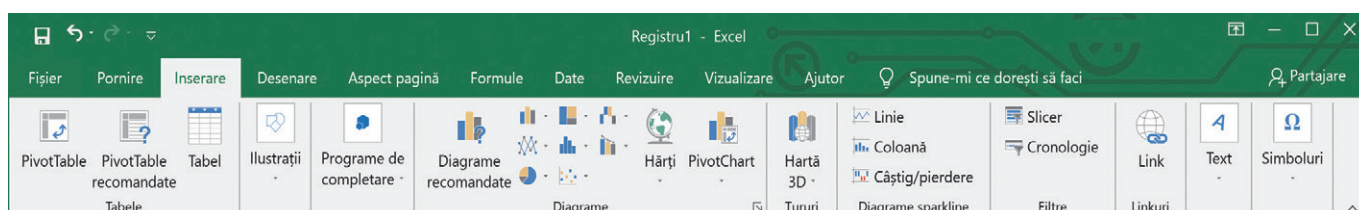
Fila **Fișier** – conține comenzile pentru acțiunile care pot fi realizate asupra fișierului MS Excel. Principalele acțiuni și cele mai uzuale sunt: crearea unui fișier nou, deschiderea unuia existent, salvarea fișierului, exportul acestuia într-un alt format, partajarea, tipărirea și închiderea sa. Tot de aici se pot gestiona opțiunile generale pentru lucrul în Excel. La selectarea acestei file, se deschide o vizualizare nouă, numită **Backstage** (imaginea alăturată).



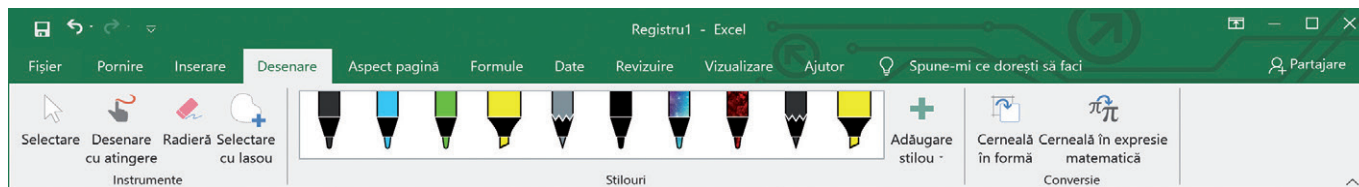
Fila Pornire – conține comenzile și elementele necesare pentru editarea și formatarea celulelor unui tabel (caracteristici litere, aliniere, stil, contururi de celule, borduri etc.).



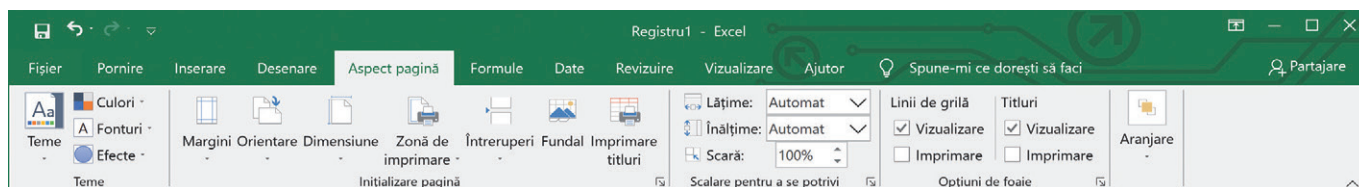
Fila **Inserare** – conține comenzile necesare pentru adăugarea altor elemente de conținut într-o foaie de calcul, de exemplu: tabele, ilustrații, diagrame, hărți, conținuturi video, legături către diferite pagini web, texte, simboluri etc.



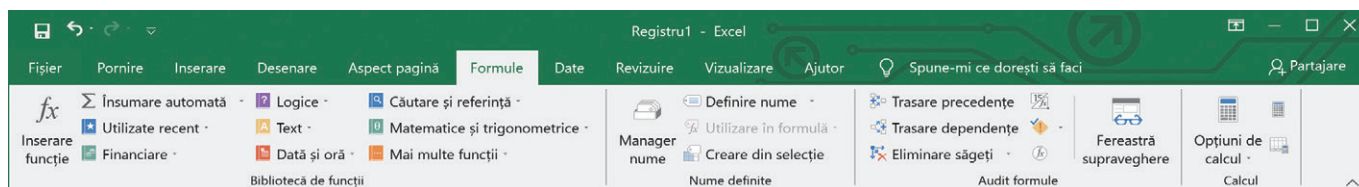
► Fila **Desenare** – conține diferite instrumente cu ajutorul cărora se pot insera în foaia de calcul notițe, sublinieri, desene, cu diferite culori, în diferite stiluri. În plus, tot din această filă se pot adăuga și expresii matematice care pot fi chiar scrise de mână, iar recunoașterea acestora se face automat de către program (butonul **Cerneală în expresie matematică**). Este activată automat pentru dispozitive cu ecrane tactile sau se poate activa de la **Fișier > Opțiuni > Particularizare panglică > Desenare**.



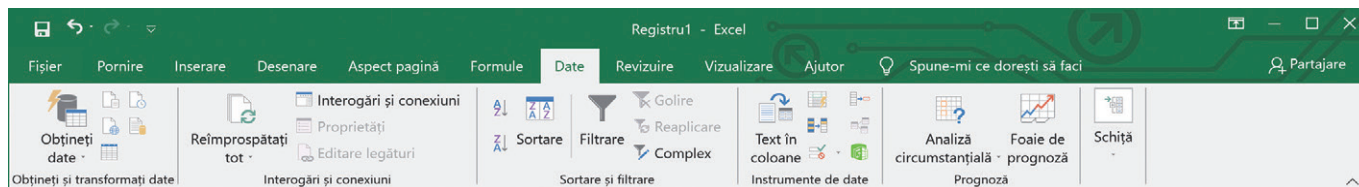
Fila **Aspect pagină** – conține comenzile pentru configurarea aspectului paginilor documentului. Aici se configurează aspectul fiecărei pagini, marginile, dimensiunea, tipul de orientare: **Portret** (vertical) sau **Vedere** (orizontal). Tot aici găsim comenzi pentru aranjarea tabelor foarte mari, cu multe celule, pentru a fi mai ușor de urmărit, dar și pentru aranjarea (poziționarea, încadrarea, alinierea) diferitelor obiecte incluse în document (de exemplu, imagini) față de celulele care au conținut (numere sau texte).



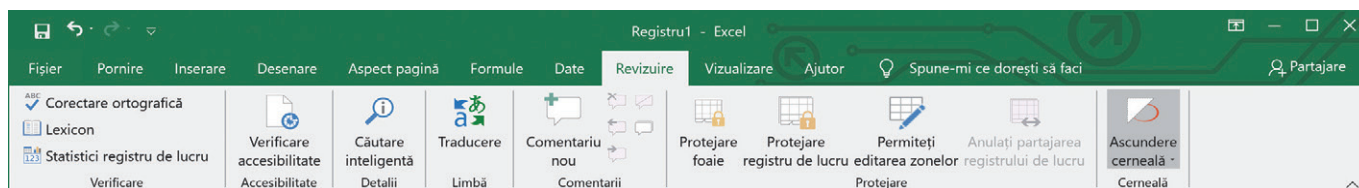
Fila **Formule** – conține comenzi pentru utilizarea diferitelor formule în foaia de calcul, de la însumare automată, până la funcții matematice complexe, trecând prin funcții logice, funcții pentru lucrul cu caractere etc. Tot aici găsim două comenzi foarte utile: **Trasare precedente** și **Trasare dependente**. Prima afișează săgeți ce indică celulele care afectează valoarea celulei selectate în prezent, iar a doua afișează săgeți care indică celulele afectate de valoarea celulei selectate în prezent.



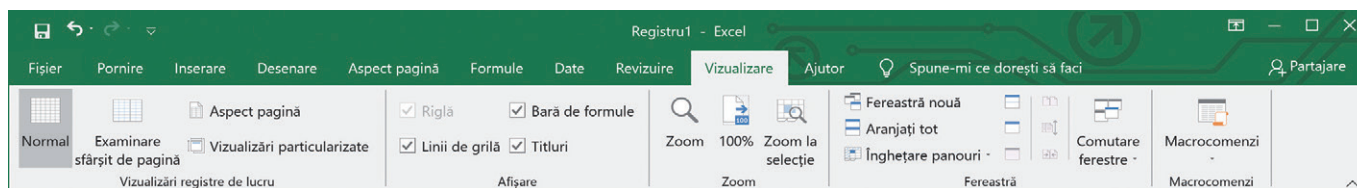
Fila **Date** – conține comenzi pentru lucrul cu datele din foaia de calcul. De exemplu, comenzi de sortare a datelor (ordonare alfabetică sau valorică, crescătoare sau descrescătoare), comenzi de filtrare a datelor din celulele selectate sau comenzi de scindare a unei coloane de text în mai multe coloane (puteți să separați o coloană de nume complete în coloane separate pentru nume și prenume) etc.



⊕ Fila **Revizuire** – permite verificarea ortografică a textelor din tabel conform cu limba în care acestea sunt scrise, adăugarea de comentarii, traducerea etc. Ea conține comenzile necesare vizualizării diferitelor statistici legate de registrul de lucru. Protejarea prin parolă a unei foi de calcul sau a unui registru este, de asemenea, posibilă din această filă.



Fila **Vizualizare** – permite alegerea modalității de vizualizare a foii de calcul în fereastră. Există aici posibilitatea de afișare în fereastra de lucru a unor instrumente ajutătoare (**Rigla**, **Linii de grilă**, **Bara de formule** sau **Titluri**) sau a unor instrumente care ajută la afișarea foii de calcul sau a unei zone selectate mărite sau micșorate.



Fila **Ajutor** – conține comenzi care permit căutarea de diferite informații de ajutor pentru utilizarea acestui produs software, precum și transmiterea de feedback firmei producătoare, pentru a sprijini îmbunătățirea pachetului de programe MS Office.

Fila **Spune-mi ce dorești să faci** – ne poate ajuta să regăsim mai rapid, în multitudinea de butoane disponibile, acele butoane care se folosesc la îndeplinirea sarcinii pe care am scris-o în căsuța de text care apare odată cu apăsarea pe numele acestei file.

Fila **Partajare** – de aici puteți partaja fișierul vostru, pentru a lucra la el împreună cu alți colegi (necesită mai întâi salvarea online în cloud).



Aplicații și investigații

Realizați următoarele acțiuni în ordinea în care sunt specificate. Observați, la fiecare pas, ce se întâmplă:

- 1 Deschideți MS Excel și creați un registru de lucru nou, necompletat; observați cum arată interfața și registrul vostru.
- 2 Scrieți pe coloana A numele tuturor colegilor de clasă. Veți observa că numele sunt mai lungi decât lungimea celei. Pentru a rezolva această mică problemă, duceți-vă cu mouse-ul pe linia dintre capetele de coloane A și B, apăsați pe ea, țineți apăsat butonul mouse-ului și trageți la dreapta până încape și cel mai lung nume din listă.
- 3 Completați în coloana B timpii realizați la sport, la proba de 50 m viteză, cu două zecimale.
- 4 Selectați toată coloana B (faceți clic pe capul de coloană), clic cu butonul drept undeva pe zona selectată, alegeți **Formatare celule**, de la **Categorie** alegeți **Număr**, iar la **Număr zecimale** puneți 2.
- 5 Completați în coloana C distanța realizată la proba de săritură în lungime de pe loc, în metri, tot cu două zecimale.
- 6 Ne dorim să afișăm tabelul ordonat alfabetic, după nume. Pentru aceasta: poziționați-vă pe oricare celulă din tabel, mergeți la fila **Date**, faceți clic pe **Sortare** (se va selecta automat zona cu celulele completate din tabel), iar din fereastra care apare, de la **Sortare după**, alegeți **Coloană A** și apăsați **OK**.
- 7 Observați că numele au apărut imediat aranjate alfabetic și, bineînțeles, în dreptul fiecărui nume s-au păstrat rezultatele obținute la cele două probe (deci odată cu celula cu numele s-au deplasat și celelalte celule de pe aceeași linie).



Exersați



- 1 Scrieți în caiete răspunsul corect:
 - a MS Excel este: • Un procesor de texte; • Un procesor de sunete; • Un program de calcul tabelar.
 - b Panglica aplicației MS Excel conține fila: • **Corespondență**; • **Tranziții**; • **Date**.
- 2 Enumerați cel puțin două moduri de vizualizare a unei foi de calcul.
- 3 Configurați nivelul de panoramare (zoom) la 55%. În câte moduri puteți modifica acest nivel?
- 4 Configurați panglica aplicației MS Excel astfel încât să fie vizibile doar filele.
- 5 Bara de titlu conține în partea stângă o zonă numită **Bara de acces rapid**. Adăugați în **Bara de acces rapid** butoanele **Deschidere** și **Imprimare rapidă**.
- 6 Scrieți în caiete, A (Adevărat) sau F (Fals), în funcție de valoarea de adevăr a propozițiilor:

a Un fișier pregătit cu ajutorul aplicației MS Excel poate avea extensia .xlsx.	A / F
b În filele aplicației MS Excel regăsim comenzi pentru afișarea statisticilor legate de foaia de calcul.	A / F
c MS Excel poate lucra numai (și obligatoriu) cu două foi de calcul într-un registru.	A / F
d În MS Excel, fila Inserare conține grupul de comenzi Diagrame .	A / F
- 7 Enumerați trei comenzi aflate pe panglica aplicației MS Excel a căror denumire începe cu litera inițială a numelui sau a prenumelui vostru.
- 8 **Dezbatere.** Organizați o dezbatere cu tema „Ordonarea (sortarea) după anumite criterii a datelor dintr-o foaie de calcul – o acțiune utilă sau împovărătoare?”.

Operații principale cu un registru de calcul



Amintiți-vă

Cu ajutorul aplicației MS Excel se pot crea **foi de calcul electronice**. Acestea pot fi salvate în format electronic pe un mediu de stocare (ca fișiere) și/sau pot fi tipărite la imprimantă.

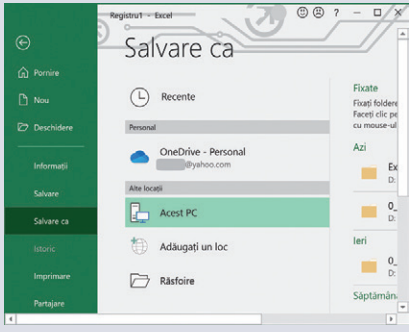
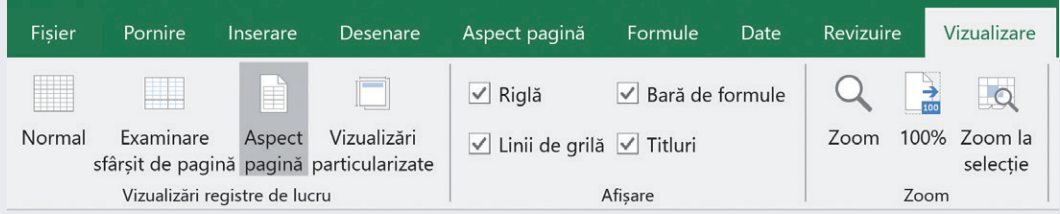
MS Excel ne pune la dispoziție o multitudine de comenzi cu ajutorul cărora putem realiza cu ușurință tabele, de la cele mai simple până la cele mai complexe, și le putem da și o formă estetică deosebită. **Panglica** include toate comenzile aplicației MS Excel grupate în file, pe tipuri de acțiuni.

Registru de calcul este fișierul creat cu ajutorul aplicației de calcul tabelar (în cazul nostru, MS Excel) și poate fi format din mai multe foi de calcul.



Descoperiți

Operații elementare de lucru cu un registru de calcul

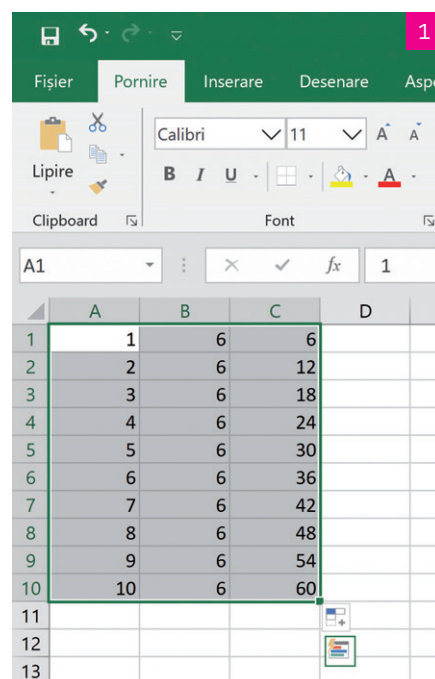
Denumirea operației efectuate asupra registrului de calcul (fișier .xlsx)	Detalierea acțiunilor efectuate în MS Excel	
Creare	Pentru a crea un registru nou, alegeți Registru de lucru necompletat din fereastra care se deschide la pornirea MS Excel. De asemenea, mai puteți apela la una dintre următoarele modalități: alegeți comanda Nou din fila Fișier sau apăsați combinația de taste Ctrl+N . Comanda Nou poate fi adăugată și în Bara de acces rapid .	
Deschidere	Pentru a deschide un registru existent, selectați comanda Deschidere din fila Fișier sau apăsați Ctrl+O . Comanda Deschidere poate fi adăugată și în Bara de acces rapid .	
Salvare	<ul style="list-style-type: none"> • Selectarea comenzii Salvare ca din fila Fișier permite salvarea fișierului curent cu un nume ales de voi, într-o locație aleasă de voi, pe calculatorul propriu, pe un dispozitiv extern (un mediu de stocare conectat la calculator) sau în cloud. • Salvarea unui fișier, fără a i se schimba numele, tipul (extensia) sau locația, se poate face selectând comanda Salvare din fila Fișier, apăsând butonul Salvare din bara de acces rapid sau combinația de taste Ctrl+S. 	
Vizualizare	<p>Accesați fila Vizualizare și alegeți opțiunea dorită (Normal, Aspect pagină sau Examinare sfârșit de pagină).</p> 	
Imprimare (tipărire)	Pentru a tipări registrul de calcul, trebuie să alegeți comanda Imprimare aflată în fila Fișier . Documentul se poate tipări la o imprimantă instalată în prealabil sau poate fi tipărit într-un fișier de tip .pdf care se salvează pe unul dintre mediile de stocare aflate la dispoziție. Acest lucru se alege din caseta Imprimantă .	
Închidere	Pentru a închide un registru deschis, selectați comanda Închidere din fila Fișier sau apăsați butonul de închidere (<i>Close</i>) din colțul din dreapta-sus al ferestrei.	

Pentru a vă familiariza cu interfața aplicației MS Excel și cu operațiile de bază pe care le puteți face cu aceasta, executați, în ordinea dată, următoarele acțiuni:

- Deschideți aplicația MS Excel. Din fila **Fișier**, deschideți submeniul **Nou** și alegeți un **Registru de lucru necompletat**.
- Scrieți tabla înmulțirii cu 6 în foaia de calcul a registrului creat anterior, astfel: pe prima coloană scrieți numerele de la 1 la 10, pe a doua coloană scrieți de 10 ori cifra 6 (în fiecare celulă din dreptul celulelor cu numerele de la 1 la 10), iar în a treia coloană scrieți rezultatele pentru fiecare înmulțire din tabla înmulțirii cu 6.
- Apăsăți combinația de taste **Ctrl+S**. Ce a apărut pe ecran?
- Salvați registrul cu numele „Tabla înmulțirii cu 6”: din fila **Fișier**, alegeți opțiunea **Salvează ca**. Alegeți locația în care doriți să salvați (de preferat un folder al vostru, de lucru) și dați-i numele precizat anterior. Apoi apăsați butonul **Salvare**.
- Din fila **Fișier**, alegeți opțiunea **Închide**.
- Acum redeschideți registrul salvat anterior folosind din fila **Fișier** opțiunea **Deschide**. În lista de fișiere **Recente** veți regăsi registrul de calcul „Tabla înmulțirii cu 6”. Clic pe el și se va deschide.

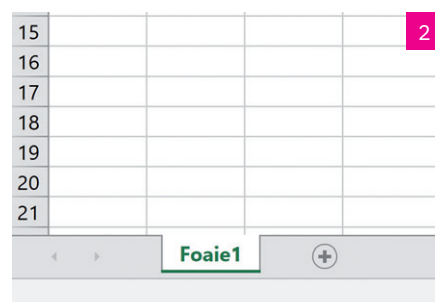
▶ În continuare, vom face câteva operații de selectare, copiere, mutare și ștergere în foaia de calcul cu tabla înmulțirii:

- Selectați întreaga tablă a înmulțirii: faceți clic în celula **A1**, țineți apăsat butonul mouse-ului și trageți până la celula **C10**. Ați selectat întreaga tablă a înmulțirii cu 6 (figura 1).
- Copiați în clipboard zona selectată pentru a o duplica în aceeași foaie de calcul. Pentru aceasta, aveți trei posibilități:
 - 1 În fila **Pornire**, mergeți la grupul de comenzi **Clipboard** și apăsați butonul **Copiere**;
 - 2 Clic cu butonul drept al mouse-ului pe zona selectată și din meniul contextual care vi se deschide alegeți opțiunea **Copiere**;
 - 3 Apăsăți combinația de taste **Ctrl+C**.
- Poziționați-vă în celula **E1** și tot de la grupul de comenzi **Clipboard** din fila **Pornire** apăsați butonul **Lipire** (sau apăsați combinația de taste **Ctrl+V**). Sau apăsați direct tasta **Enter**. Acum aveți pe foaia de calcul de două ori tabla înmulțirii cu 6.
- Dacă doriți să mutați zona selectată și nu să o copiați, în loc să alegeți **Copiere** din **Clipboard**, alegeți **Decupare** (sau apăsați combinația de taste **Ctrl+X**), apoi poziționați-vă în locul unde vreți să o mutați și apăsați butonul **Lipire** sau apăsați tasta **Enter**. Faceți acest lucru cu a doua tablă a înmulțirii din foaia voastră de calcul.
- Acum ștergeți cea de-a doua tablă a înmulțirii din foaia de calcul: selectați-o și apăsați tasta **Delete** sau apăsați combinația de taste **Ctrl+X** sau butonul **Decupare** din fila **Pornire**. Dacă veți folosi una din ultimele două variante, se va șterge tabla înmulțirii selectată din foaia voastră de calcul, dar va rămâne în memorie, în clipboard, și o veți putea „lipi” mai târziu în altă parte.

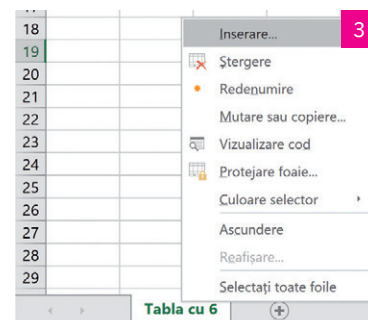


Operații cu foi de calcul într-un registru: creare (inserare), accesare, redenumire, ștergere

- Observați în partea stângă-jos a aplicației eticheta cu numele foii de calcul create automat la crearea registrului. Implicit poartă numele „Foaie1” (figura 2).
- Pentru a-i schimba numele, faceți dublu clic pe etichetă și scrieți noul nume: „Tabla cu 6”.
- Apăsăți butonul **+** de lângă eticheta cu numele foii de calcul inițiale și creați (inserați) două noi foi de calcul.
- Modificați numele noilor foi de calcul în „Tabla cu 8”, respectiv „Tabla cu 9”.
- Copiați tabla înmulțirii cu 6 în cele două foi de calcul nou create și faceți modificările, astfel încât fiecare foaie să conțină tabla înmulțirii corespunzătoare numelui deținut.
- Pentru a accesa o foaie de calcul sau alta, doar faceți clic pe eticheta cu numele foii dorite. Faceți clic pe eticheta cu numele „Tabla cu 8” și respectiva foaie va trece în față și va putea fi accesată pentru vizualizare, modificare etc.
- Dacă doriți să schimbați ordinea foilor de calcul, de exemplu, ca prima (de la stânga la dreapta) foaie să fie „Tabla cu 9”, trageți, folosind **drag & drop**, eticheta cu numele „Tabla cu 9” înaintea celorlalte două; astfel ajung în ordinea dorită.
- Ștergeți foaia de calcul „Tabla cu 8”: faceți clic cu butonul drept al mouse-ului pe eticheta cu numele „Tabla cu 8” și selectați opțiunea **Ștergere**.



După cum puteți observa în imaginea alăturată (figura 3), acțiunile de **Inserare** (creare), **Ștergere**, **Redenumire** și **Mutare sau copiere...** sunt disponibile, alături de alte câteva, în meniul contextual care apare la un clic de buton drept al mouse-ului pe eticheta cu numele unei foi de calcul.



- La apăsarea butonului **Mutare sau copiere...**, se deschide o fereastră din care se alege:
- 1 în care registru se dorește copierea/mutarea (câmpul **În registrul**);
 - 2 înaintea căreia foi de calcul se introduce fila respectivă (se alege din listă **Înainte de foii**);
 - 3 dacă se dorește mutarea sau copierea; în acest ultim caz se bifează **Crearea unei copii**.



Aplicați și investigați

Realizați un tabel de colectare a datelor (prin introducere directă) pentru un studiu statistic simplu.

Pentru aceasta, creați un registru Excel și salvați-l în cloud cu numele „Timp la televizor.xlsx”. Trimiteți invitația de a lucra pe acest registru cât mai multor colegi din clasa voastră, de preferat tuturor.

Faceți următorul experiment: urmăriți-vă (toți colegii implicați trebuie să facă același lucru) timp de o săptămână și notați în fiecare zi câte minute vă uitați la televizor, indiferent la ce emisiune. Notați datele în registrul „Timp la televizor.xlsx”, într-un tabel similar cu cel de mai jos. În loc de „Nume 1”, „Nume 2”, fiecare participant la studiu își va trece numele propriu.

Pentru ca tabelul să și arate frumos, selectați celulele cu zilele săptămânii și, de la butonul **Culoare de umplere** (butonul cu gălețușa de vopsea) din grupul **Font**, fila **Formatare**, alegeți o nuanță deschisă de portocaliu, apoi selectați celulele cu numele participanților la studiu și colorați-le într-o nuanță de albastru-deschis.

Selectați întregul tabel și tot de la grupul **Font**, de la butonul **Borduri**, alegeți **toate bordurile** pentru a marca marginile tuturor celulelor cu o linie.

	Luni	Marti	Miercuri	Joi	Vineri	Sambata	Duminica
Nume 1							
Nume 2							
Nume 3							
Nume 4							
Nume 5							
Nume 6							
Nume 7							
Nume 8							
Nume 9							
Nume 10							

La finalul săptămânii, adunați minutele din toate cele șapte zile pentru fiecare nume în parte și gândiți-vă câte lucruri ați fi putut face în plus în acest timp petrecut la televizor.



Exersați



- 1 Asociați corect noțiunile referitoare la MS Excel din cele două coloane:

a. Ctrl+N	I. Afișarea mărită/micșorată a unei foi de calcul
b. Aspect pagină	II. Creare de nou registru de calcul
c. Zoom	III. Mod de vizualizare a unui registru de lucru
- 2 Care este rezultatul accesării comenzii **Salvare ca** din fila **Fișier**?
- 3 Alegeți variantele corecte.
 - I. Extensia unui fișier creat în MS Excel poate fi:

a xlsx;	b docx;	c mp4.
---------	---------	--------
 - II. Alegerea unui mod de vizualizare se realizează accesând:
 - a grupul de comenzi **Afișare** din fila **Vizualizare**;
 - b grupul de comenzi **Vizualizări registre de lucru** din fila **Vizualizare**;
 - c grupul de comenzi **Fereastră** din fila **Vizualizare**.
- 4 **Portofoliu.** Alcătuiți un eseu de maximum o pagină cu tema „Ce s-ar întâmpla dacă ați uita să salvați un fișier la care ați lucrat singur, într-o duminică, timp de 7 ore, termenul de predare fiind în două zile?”. Includeți în eseu propuneri de acțiuni atât pentru remedierea problemei, cât și proactive, de preîntâmpinare a unui astfel de eveniment.

Formatarea unui registru de calcul. Tipuri de date. Formule



Amintiți-vă



Am văzut în lecția anterioară că putem aranja tabelele și din punct de vedere estetic. Puteți colora unele celule ale foii de calcul: doar selectați celulele dorite și, de la butonul **Culoare de umplere** (butonul cu gălețușa de vopsea) din grupul **Font**, fila **Pornire**, alegeți o culoare. Puteți pune linii care să marcheze coloanele și liniile din foaia de calcul: selectați zona dorită sau întregul tabel și tot de la grupul **Font**, de la butonul **Borduri**, alegeți **Toate bordurile** pentru a marca marginile tuturor celulelor cu o linie.

Se pot face însă și alte operații de formatare a rândurilor și coloanelor unei foi de calcul, precum și diverse operații de formatare a celulelor, în funcție de informația pe care o conțin.



Descoperiți

1 Noțiuni de formatare

► a Operații de formatare a rândurilor/coloanelor

Să reluăm registrul de calcul cu tabla înmulțirii cu 6.

În celulele completate cu tabla înmulțirii, spațiul ocupat de numere este mic, în comparație cu dimensiunea celulelor. Cum facem să micșorăm dimensiunea celulelor, astfel încât întreaga tablă a înmulțirii să ocupe un spațiu mai mic? Poziționați mouse-ul pe linia dintre literele care etichetează două coloane alăturate (de exemplu, între **A** și **B**). Cursorul mouse-ului se va transforma într-o linie verticală cu două săgețe spre stânga și spre dreapta, sugerând astfel că puteți trage de acea linie într-o direcție sau alta, micșorând, respectiv măbind lățimea întregii coloane din stânga cursorului, în cazul nostru coloana **A**. Țineți butonul stâng al mouse-ului apăsat și trageți spre stânga, micșorând lățimea coloanei cât credeți de cuviință. Procedați la fel cu coloanele **B** și **C**. Dacă veți face dublu clic în momentul în care cursorul mouse-ului este transformat după cum am văzut mai sus, lățimea coloanei se va ajusta automat la mărimea minimă, astfel încât să încapă textul cel mai lung din celulele coloanei respective (figurile 1 și 2).

Procedați similar pentru ajustarea dimensiunii verticale a unui rând. Se selectează rândurile pentru care se dorește modificarea dimensiunii, se poziționează cursorul mouse-ului pe linia dintre cifrele care constituie etichetele rândurilor și când acesta se transformă într-o linie verticală cu două săgeți în sus și în jos, se trage în direcția necesară, cu butonul mouse-ului apăsat, până se obține dimensiunea dorită (figura 3).

	A	B	C	D
1	1	6	6	
2	2	6	12	
3	3	6	18	
4	4	6	24	
5	5	6	30	
6	6	6	36	
7	7	6	42	
8	8	6	48	
9	9	6	54	
10	10	6	60	
11				

1. Coloanele cu dimensiunile implicite

	A	B	C	D
1	1	6	6	
2	2	6	12	
3	3	6	18	
4	4	6	24	
5	5	6	30	
6	6	6	36	
7	7	6	42	
8	8	6	48	
9	9	6	54	
10	10	6	60	
11				

2. Coloanele cu dimensiunile micșorate

	A	B	C	D
1	1	6	6	
2	2	6	12	
3	3	6	18	
4	4	6	24	
5	5	6	30	
6	6	6	36	
7	7	6	42	
8	8	6	48	
9	9	6	54	
10	10	6	60	
11				

3. Înălțimea rândurilor mărită



Rețineți

Dimensiunile unei celule pot fi ajustate atât pe orizontală, cât și pe verticală.

b Operații de formatare a celulelor

➕ Pentru o formatare ușoară și de calitate, MS Excel pune la dispoziție mai multe **stiluri de celule** predefinite pe care le puteți utiliza. Un **stil** reprezintă de fapt un set de caracteristici de formatare (fonturi, dimensiuni de font, de borduri și culori de celule etc.).

Puteți alege un astfel de stil acesând fila **Pornire** > grupul **Stiluri** > **Stiluri de celule**. Puteți să vă creați însă propria formatare, aplicând pe conținutul celulelor unei foi de calcul toate opțiunile de formatare de font și de paragraf pe care le cunoașteți de la editorul de texte. Vedeți grupurile de comenzi **Font** și **Aliniere** din fila **Pornire**.

Să aliniem centrat, pe mijlocul coloanei, și să apară îngroșate rezultatele înmulțirilor. Mai întâi, pentru a vedea mai bine rezultatul acțiunii, lățim puțin coloana C. Apoi selectăm celulele de la **C1** la **C10** (sau, mai rapid, facem clic pe litera C, care etichetează coloana, selectând-o astfel) și facem clic pe butonul **Aldin** din grupul **Font** și pe butonul **La centru** din grupul **Aliniere** (figura 4).

	A	B	C	D
1	1	6	6	
2	2	6	12	
3	3	6	18	
4	4	6	24	
5	5	6	30	
6	6	6	36	
7	7	6	42	
8	8	6	48	
9	9	6	54	
10	10	6	60	
11				

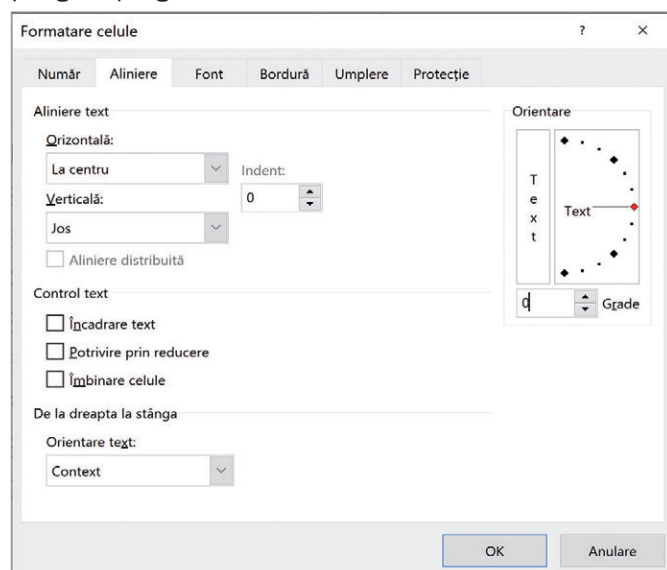
4. Coloana C – îngroșată și centrată



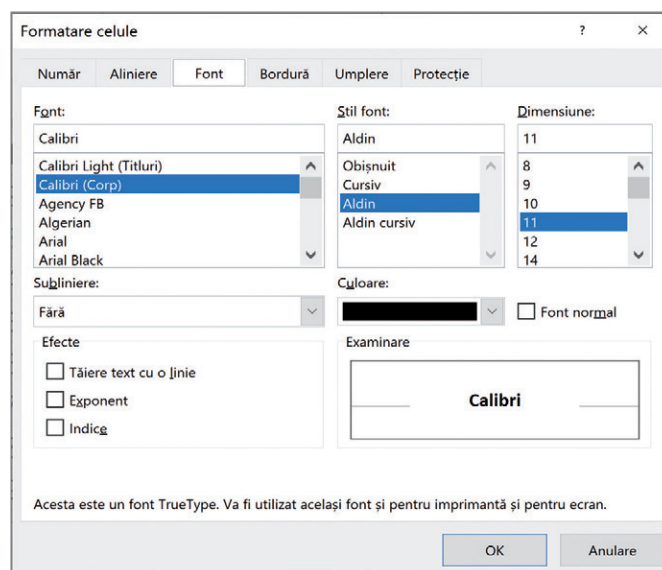
Rețineți

Asupra unei celule pot fi aplicate mai multe **opțiuni de formatare**. Pentru a vedea toate posibilitățile de formatare disponibile, faceți clic cu butonul drept al mouse-ului pe o celulă, iar din meniul contextual care se deschide alegeți **Formatare celule...**. Va apărea o fereastră cu mai multe file, dintre care, pentru moment, ne interesează filele **Aliniere**, **Font**, **Bordură** și **Umplere**.

La filele **Aliniere** și **Font** puteți ajunge și făcând clic pe săgețele din dreapta-jos, în grupurile cu aceleași nume din panglica programului MS Excel.



5. Fila Aliniere



6. Fila Font

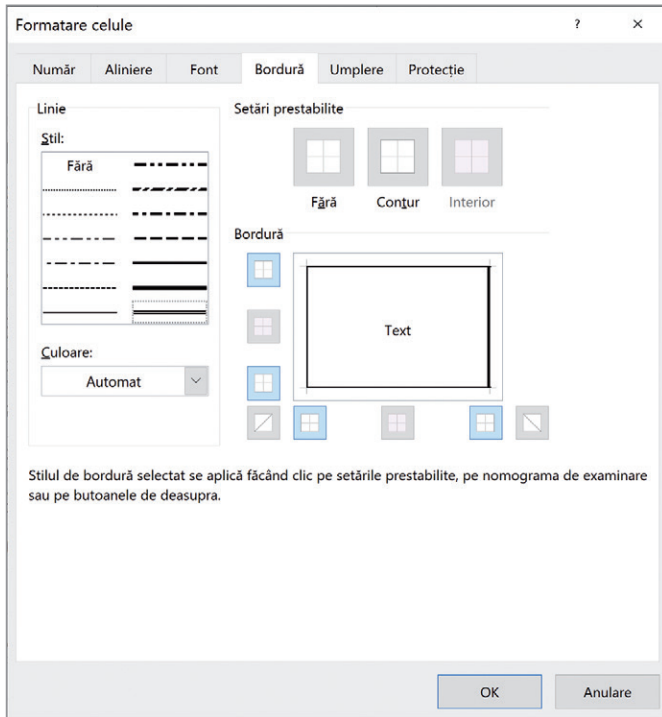
Selectați coloana A din foaia de calcul cu tabla înmulțirii și aplicați, pe rând, toate opțiunile disponibile în filele **Aliniere** și **Font**. Observați ce efect are asupra textelor din celule fiecare modificare aplicată.

Puteți alinia textul pe orizontală sau pe verticală alegând opțiunea dorită din lista derulantă pusă la dispoziție (la stânga, la centru, la dreapta etc.). Puteți controla cum să apară textul dintr-o celulă:

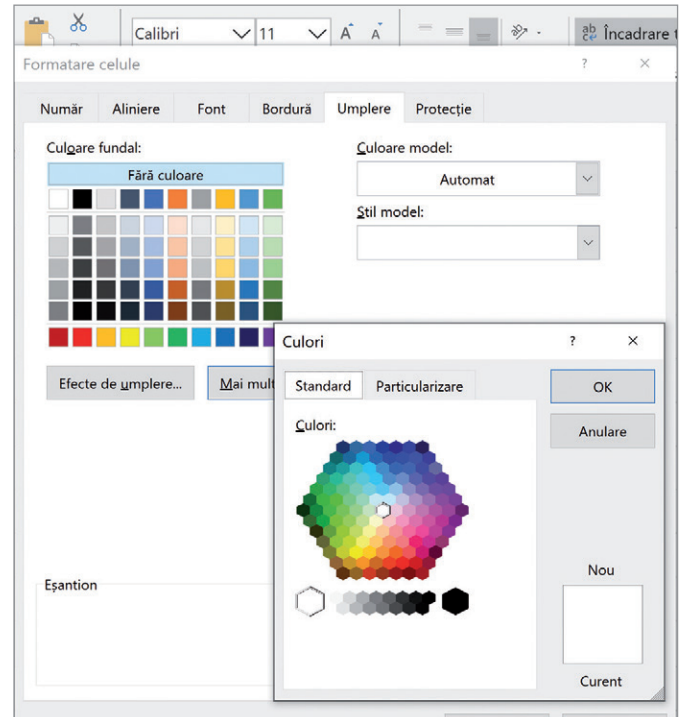
- 1 dacă dimensiunea sa e mai mare decât cea a celulei, să fie împărțit pe mai multe linii (**Încadrare text**);
- 2 dacă dimensiunea sa e mai mare decât cea a celulei, să fie redusă dimensiunea fontului astfel încât să se încadreze în celulă (**Potrivire prin reducere**);
- 3 să se îmbine mai multe celule selectate în prealabil, astfel încât textul să apară comun pentru toate (**Îmbinare celule**).

O opțiune interesantă este aceea de **Orientare** a textului în celule. De aici puteți configura textul să fie orizontal, vertical sau oblic (cu unghiuri specificate în grade) în celula foii de calcul.

Configurarea fontului este asemănătoare cu cea învățată la Microsoft Word. Se pot schimba fontul, stilul acestuia (obișnuit, cursiv, aldin, aldin cursiv), dimensiunea, culoarea. De asemenea, poate fi subliniat sau poate avea alte efecte (tăiat cu o linie, micșorat ca exponent sau ca indice).



7. Fila Bordură



8. Fila Umplere

+ Selectați toate celulele cu tabla înmulțirii și aplicați, pe rând, toate opțiunile disponibile în filele **Bordură** și **Umplere**. Experimentați diverse stiluri de linie pentru contururile celulelor, diferite culori de umplere pentru fundalul celulelor și diferite culori pentru scrisul din interiorul celulelor.

Observați că, deși aveți la dispoziție o multitudine de culori și nuanțe pentru umplerea fundalurilor celulelor, există și un buton **Mai multe culori...** de la care puteți alege dintre mai multe nuanțe disponibile.



Aplicați și investigați

Selectați toată tabla înmulțirii cu 6 și, de la opțiunile din fila **Borduri**, găsiți soluția pentru a avea conturul exterior al tabelului cu tabla înmulțirii marcat cu o dungă dublă, iar liniile interioare să fie marcate cu o linie punctată.

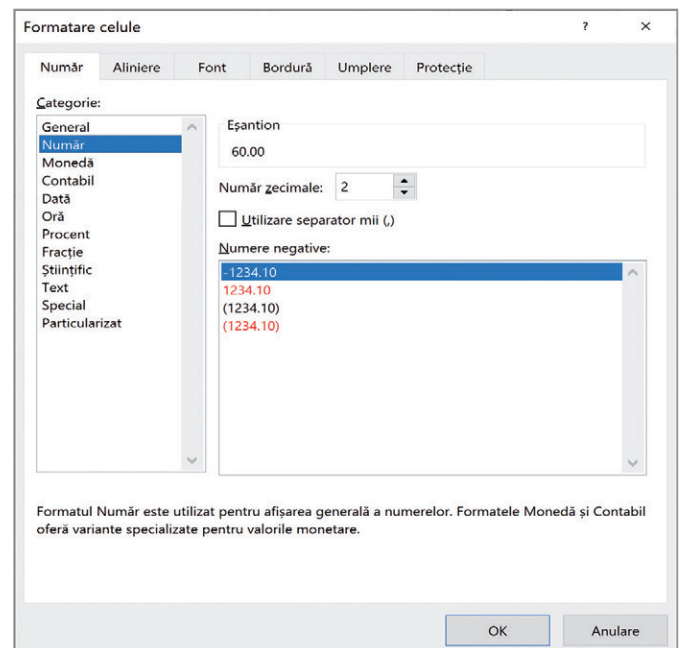


Descoperiți

2 Tipuri de date: numeric, text, dată calendaristică și altele

Poate că cea mai importantă opțiune de formatare a unei celule (din punct de vedere practic) este faptul că o putem configura pentru a avea conținutul de un anumit tip. De exemplu:

- număr (puteți configura utilizarea separatorului de mii, precum și numărul de zecimale care să se afișeze);
- monedă (puteți configura diferite monezi, inclusiv RON);
- dată sau oră (puteți face operații cu date, respectiv ore);
- procent;
- fracție (dacă selectați „fracție” și introduceți în acea celulă valoarea „5.5”, va apărea „5 1/2”);
- text etc.



3 Utilizarea formulelor de calcul într-o foaie de calcul

Menirea principală a unei foi de calcul, dincolo de a ne oferi un tabel predefinit în care putem introduce date, este aceea de a ne oferi posibilitatea de a obține rezultate calculate automat, în baza unor formule.

Să luăm exemplul foi de calcul cu tabla înmulțirii cu 6. În celula **C1**, în loc să scrieți rezultatul înmulțirii, calculat de voi, scrieți formula care calculează acel rezultat și îl afișează în celula tabelului.

O formulă într-o celulă de foaie de calcul începe întotdeauna cu semnul = . Așadar, formula noastră este: **=1*6**.



Rețineți

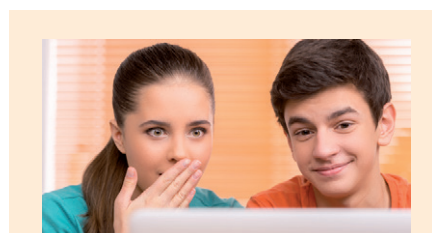
Puteți folosi toți operatorii pe care îi cunoașteți: + (adunare), - (scădere), * (înmulțire), / (împărțire), ^ (ridicare la putere) etc. Bineînțeles că se pot folosi și parantezele, cu același rol și aceeași prioritate pe care le știți de la matematică.

O facilitare și mai interesantă oferită de programele de calcul tabelar este să înmulțim nu două numere, ci valorile conținute în două celule, oricare ar fi acele valori. Astfel: pentru a obține în celula **C1** rezultatul înmulțirii lui 1 cu 6, adică ceea ce este în celula **A1** înmulțit cu ceea ce este în celula **B1**, trebuie să scriem în celula **C1** formula:

= A1*B1.

Acum, dacă vă veți poziționa în celula **C1** și o veți copia în clipboard (**Ctrl+C**), iar apoi vă veți duce în celula **C2** și veți „lipi“ (**Ctrl+V**) acolo formula copiată, veți avea surpriza ca după ce apăsați **Ctrl+V** să vă apară „12“, adică rezultatul înmulțirii lui 2 cu 6. Aceasta pentru că programul a știut să copieze și să adapteze formula de calcul astfel: dacă în **C1** s-a introdus formula **A1*B1**, atunci în **C2** se va introduce formula **A2*B2**, adică **2*6**.

Pentru a ne ușura și mai mult munca, programul mai oferă o facilitare de lucru: după ce ați introdus în celula **C1** formula de calcul cu adresare la celule (**=A1*B1**), este suficient să vă poziționați cu mouse-ul pe celula **C1** și să trageți de pătrățica din colțul din dreapta-jos al celulei până la **C10** (adică pentru toată tabla înmulțirii). Vor apărea instant rezultatele tuturor înmulțirilor cu 6, de la 1 la 10. Aceasta pentru că s-a replicat automat formula de calcul din celula **C1**, cu referință la rândul pe care apare (în **C2**, se calculează **A2*B2**, în **C3** se calculează **A3*B3**, în **C4** se calculează **A4*B4**, și așa mai departe).



ȘTIAȚI CĂ...?

Dacă introduceți un număr într-o celulă (de exemplu, numărul 8 în celula **B1**), iar apoi trageți cu mouse-ul de colțul din dreapta-jos al celulei în jos pe coloană, veți replica numărul 8 (adică ce e conținut în celula **B1**) în toate celulele peste care trageți mouse-ul. Veți obține automat o serie de date cu aceeași valoare.



Aplicați și investigați

În foaia de calcul cu tabla înmulțirii cu 6 înlocuiți, în coloana **B**, cifra 6 cu cifra 8, din celula **B1** până în celula **B10**. Ce s-a întâmplat? Rezultatele din coloana **C** au fost automat modificate și ați obținut tabla înmulțirii cu 8.



Descoperiți



4 Sortarea crescătoare/descrescătoare a datelor dintr-un tabel după unul sau mai multe criterii

În lecția anterioară ați creat registrul „Timp la televizor.xlsx“. Să presupunem că îl avem completat și arată ca în imaginea de mai jos. Acum sarcina este să puneți în ordine crescătoare, după timpul petrecut la TV în ziua de luni, toate numele din tabel. Pentru aceasta, poziționați-vă pe o celulă din tabel și urmați pașii:

- Mergeți în fila **Date**;
- Din grupul **Sortare și filtrare** apăsați butonul **Sortare**;
- În fereastra care se deschide, în câmpul **Sortare după** alegeți coloana **Luni** (sarcina era să se ordoneze după timpul petrecut la TV în ziua de luni), iar la **Ordine** alegeți **De la cel mai mic la cel mai mare** (în ordine crescătoare);
- Apăsați butonul **OK**.

	A	B	C	D	E	F	G	H
1		Luni	Marti	Miercuri	Joi	Vineri	Sambata	Duminica
2	Popescu Dan	18	233	41	33	43	0	0
3	Vrânceanu Vlad	23	43	22	51	11	0	0
4	Mihai Daniel	43	55	12	15	19	45	0
5	Isculescu Flavia	44	22	56	10	10	110	0
6	Popescu Ana	18	43	82	19	14	90	0
7	Lazăr Andrei	45	12	112	12	15	0	0
8	Popescu Virginia	18	21	23	18	9	0	0
9	Vasilescu Marian	12	14	33	15	11	0	0
10	Dănescu Andreea	5	16	22	13	23	0	0
11	Avram Sofia	33	19	11	12	21	0	0

În acest moment, veți vedea că tabelul vostru s-a reordonat după criteriile selectate, dar aveți o mică problemă: există trei colegi cu același număr de minute petrecute la TV în ziua de luni (Popescu Dan, Popescu Ana și Popescu Virginia). Pe ei cum îi ordonăm? Răspunsul firesc este: alfabetic. Adică, în caz de egalitate, vom introduce un al doilea criteriu de ordonare, care să îi ordoneze pe cei trei (în cazul nostru) care au timpi egali, după criteriul „ordonare alfabetică“.

Procedați astfel, fiind poziționați în tabel:

- Apăsați din nou butonul **Sortare**;
- Din fereastra **Sortare** apăsați butonul **Adăugare nivel**;
- În câmpul **Apoi după** alegeți **Coloană A**;
- Apăsați butonul **OK**.

Acum, tabelul, sortat după cele două criterii, arată ca în imaginea alăturată.

	A	B	C	D	E	F	G	H
1		Luni	Marti	Miercuri	Joi	Vineri	Sambata	Duminica
2	Dănescu Andreea	5	16	22	13	23	0	0
3	Vasilescu Marian	12	14	33	15	11	0	0
4	Popescu Ana	18	43	82	19	14	90	0
5	Popescu Dan	18	233	41	33	43	0	0
6	Popescu Virginia	18	21	23	18	9	0	0
7	Vrânceanu Vlad	23	43	22	51	11	0	0
8	Avram Sofia	33	19	11	12	21	0	0
9	Mihai Daniel	43	55	12	15	19	45	0
10	Iscrulescu Flavia	44	22	56	10	10	110	0
11	Lazăr Andrei	45	12	112	12	15	0	0

Observați că tot tabelul este ordonat crescător după timpii petrecuți la TV în ziua de luni, iar cei trei colegi care au timpi egali au fost ordonați alfabetic (și cum aveau inclusiv numele de familie identic, ordonarea s-a făcut după prenume: Ana, Dan și Virginia).



Rețineți

Puteți sorta (ordona) tabelul vostru în funcție de mai multe criterii de sortare odată, fiecare criteriu fiind specificat pe un nivel de sortare. Ordonarea se va realiza ținându-se cont de ordinea în care apar nivelurile de sortare, de sus în jos.

ȘTIAȚI CĂ...?

Dacă introduceți un număr într-o celulă (de exemplu, numărul 1 în celula A1), iar apoi trageți cu mouse-ul de colțul din dreapta-jos al celulei în jos pe coloană, ținând apăsată în acest timp și tasta **Ctrl**, veți obține în fiecare celulă peste care trageți mouse-ul o valoare cu 1 mai mare decât în celula anterioară. Adică în celula A2 veți obține automat valoarea 2, în celula A3, valoarea 3 și așa mai departe. Se va realiza astfel completarea automată a unei **serii de date** cu valori crescătoare.



Aplicați și investigați

Refaceți ordonarea tabelului, tot după timpii petrecuți la TV în ziua de luni, dar de această dată în ordine descrescătoare.



Exersați

- 1 Creați o foaie de calcul, completați un tabel cu 4 coloane și 6 rânduri cu numere diferite, atât întregi, cât și cu zecimale. Formatați tabelul, astfel încât să afișeze toate numerele cu separatorul de mii și cu trei zecimale.
- 2 De ce tip sunt datele din celulele unui registru, dacă în două dintre celule avem valorile 14:55 și 13:30, iar în a treia celulă rezultatul scăderii lor este 1:25?
 - a Număr;
 - b Oră;
 - c Contabil.
- 3 Realizați în maximum 3 minute o foaie de calcul cu tablele înmulțirii cu 6, 7, 8, 9 și 10, cu înmulțiri de la 1 la 50 pentru fiecare. Folosiți formule de calcul și multiplicarea conținutului unei celule, cu și fără tasta **Ctrl**, în funcție de necesități.
- 4 Scrieți, în caiete, **A** (Adevărat) sau **F** (Fals) pentru fiecare propoziție de mai jos:
 - a Textul dintr-o celulă poate fi formatat să apară chiar și pe verticală. A / F
 - b Nu se pot formata caracterele dintr-o celulă cu fonturi și mărimi diferite. A / F
 - c Scrierea unei formule de calcul începe întotdeauna cu semnul = . A / F
- 5 **Dezbateri.** Inițiați o dezbateri în clasă pe tema „Adăugarea de formule într-un registru de calcul înseamnă automatizarea calculelor și eficiență sporită sau doar o complicație în plus?“.
- 6 **Portofoliu.** Realizați în MS Excel un carnet de note electronic, cu toate materiile pe care le faceți la școală. Puneți coloane pentru note, coloane separate pentru fiecare semestru, pentru teze, pentru medii și calcularea mediei generale. Atenție la formulele de calcul ale mediilor pentru materiile la care aveți și teze.

Modalități de utilizare a registrelor de calcul. Calcule tabelare, funcții



Descoperiți

Am văzut că celulele unei foi de calcul pot conține date de diferite tipuri (număr, text, monedă, dată, oră, procent, fracție etc.) și că putem folosi formule într-o foaie de calcul pentru ușurarea lucrului. Vom vedea împreună câteva modalități practice de utilizare a foilor de calcul și câteva funcții specifice aplicației de calcul tabelar.

Formule de calcul care utilizează operatori aritmetici (+, -, *, /)

Să pornim de la tabelul alăturat.

- Pentru a calcula suma în celula C2, folosim operatorul aritmetic +, iar formula este: **=87+33**.
Dacă vrem (ceea ce este de preferat) să folosim referințe de celule, formula este: **=A2+B2**.
- Pentru a calcula diferența în celula C5, folosim operatorul aritmetic -, iar formula este: **=100-53**.
Dacă folosim referințe de celule, formula este: **=A5-B5**.
- Pentru a calcula produsul în celula C8, folosim operatorul aritmetic *, iar formula este: **=9*8**.
Dacă folosim referințe de celule, formula este: **=A8*B8**.
- Pentru a calcula câtul în celula C11, folosim operatorul aritmetic /, iar formula este: **=99/9**.
Dacă folosim referințe de celule, formula este: **=A11/B11**.

	A	B	C
1	Primul nr	Al 2-lea nr	SUMA
2	87	33	
3			
4	Primul nr	Al 2-lea nr	DIFERENȚA
5	100	53	
6			
7	Primul nr	Al 2-lea nr	PRODUSUL
8	9	8	
9			
10	Primul nr	Al 2-lea nr	CÂȚUL
11	99	9	



Rețineți

Puteți folosi operatorii aritmetici cunoscuți pentru a obține expresii oricât de complexe legate de conținutul celulelor. De exemplu: **=(A1+A2)*(A4-B4)/((A8*B8)+(A11/B11))**.



Funcții specifice aplicației de calcul tabelar pentru sumă, maxim, minim, medie aritmetică și decizie



- Funcția SUM – SUMĂ**
Descriere: adună valori. Puteți aduna valori individuale, referințe de celule, zone sau o combinație a acestora.
Sintaxa este: **=SUM(număr1, [număr2], ...)**.
De exemplu: **=SUM(A2:A10)** adună valorile din toate celulele de la A2 la A10.
=SUM(A2:A10, C2:C10) adună valorile din celulele de la A2 la A10, precum și valorile din celulele de la C2 la C10.
- Funcția MAX – MAXIM**
Descriere: întoarce valoarea cea mai mare dintr-un set de valori.
Sintaxa este: **=MAX(număr1, [număr2], ...)**.
De exemplu: **=MAX(4, 99, 43)** întoarce valoarea 99.
=MAX(A2:A10) afișează maximul celulelor de la A2 la A10.
- Funcția MIN – MINIM**
Descriere: întoarce valoarea cea mai mică dintr-un set de valori.
Sintaxa este: **=MIN(număr1, [număr2], ...)**.
De exemplu: **=MIN(4, 99, 43)** întoarce valoarea 4.
=MIN(A2:A10) afișează minimul celulelor de la A2 la A10.
- Funcția AVERAGE – MEDIE ARITMETICĂ**
Descriere: calculează media aritmetică a unui grup de numere.
Sintaxa este: **=AVERAGE (număr1, [număr2], ...)**.
De exemplu: **=AVERAGE (10, 2, 18)** întoarce valoarea 10.
=AVERAGE(A2:A10) afișează media aritmetică a numerelor corespunzătoare celulelor de la A2 la A10.

5 Funcția IF – DACĂ

Descriere: este funcția de decizie. Este una dintre cele mai populare funcții din Excel și vă permite să efectuați comparații logice între o valoare și valoarea la care vă așteptați.

Așadar, o funcție IF poate avea două rezultate. Primul rezultat apare când comparația este TRUE (ADEVĂRAT), al doilea când este FALSE (FALS).

Sintaxa este: =IF (test_logic, [valoare_pt_ADEVĂRAT], [valoare_pt_FALS])

De exemplu: în celula D4 din tabelul alăturat avem calculat un punctaj la un examen. Condiția pentru a promova examenul este ca punctajul să fie cel puțin 80.

=IF(D4>=80, „PROMOVAT“, „RESPINS“) întoarce rezultatul „PROMOVAT“, dacă punctajul calculat în celula D4 este mai mare sau egal cu 80, dar întoarce „RESPINS“, dacă punctajul este mai mic de 80.

	A	B	C	D	E
		Punctaj 1	Punctaj 2	TOTAL	
1	NUME, PRENUME				PROMOVAT/RESPINS
2	Popescu Dan	45	33	78	RESPINS
3	Vrânceanu Vlad	45	46	91	PROMOVAT
4	Mihai Daniel	42	41	83	PROMOVAT
5	Iscrulescu Flavia	39	40	79	RESPINS
6	Popescu Ana	38	22	60	RESPINS
7	Lazăr Andrei	45	55	100	PROMOVAT
8	Popescu Virginia	44	44	88	PROMOVAT
9	Vasilescu Marian	39	47	86	PROMOVAT
10	Dănescu Andreea	38	37	75	RESPINS
11	Avram Sofia	48	49	97	PROMOVAT

 **Rețineți**

Puteți utiliza toate aceste funcții împreună cu operatorii aritmetici învățați pentru a crea expresii mai complexe. De exemplu: =Sum(B1,B2)*MIN(C1,C2,C3)/MAX(C4,C5)/IF(C11=49,2,3).

 **Aplicați și investigați**

1 Faceți un tabel cu o listă de cumpărături (în coloana A) și prețurile lor în Euro (în coloana B).

Introduceți (să zicem, în celula K1) cursul Leu/Euro. Faceți și o coloană (C) cu prețurile în lei, calculate în funcție de cursul Euro. Efectuați următoarele analize: cât s-a cheltuit în total (suma), în coloana D; care este produsul cel mai ieftin (minimumul), în coloana E; care este produsul cel mai scump (maximumul), în coloana F; care este prețul mediu (media aritmetică), în coloana G; dacă suma depășește valoarea de 10 000 lei, afișați mesajul „BUGET DEPĂȘIT“, altfel afișați mesajul „V-AȚI ÎNCADRAT ÎN BUGET“ (decizie).

La final, calculați ce sumă de bani din ceea ce s-a cheltuit merge la stat (care este valoarea TVA-ului pentru cumpărăturile efectuate)? TVA-ul reprezintă „Taxa pe Valoarea Adăugată“ și este de 19%. Pentru ușurința calculului, vom proceda astfel: în coloana H vom calcula prețul produselor fără TVA (în lei), după următoarea formulă:

=preț/1.19 (concret, în tabelul vostru introduceți în celula H2 formula =C2/1.19 și replicați această formulă pe coloana H pentru toate produsele din tabel).

Apoi, în coloana I, calculați valoarea TVA-ului (prețul dat în lei minus prețul fără TVA, cel calculat în coloana H). Faceți acest calcul pentru toate produsele, iar la final calculați totalul TVA-urilor.

2 Putem utiliza foile de calcul cu funcții și formule pentru rezolvarea unor probleme la fizică, matematică, chimie, geografie etc.? Răspunsul este DA. Să vedem câteva exemple.

2.1. Rezolvarea unei probleme de fizică

Să luăm următoarea problemă dintr-o culegere de fizică: *Într-un experiment de monitorizare a balenelor, biologi au constatat că o balenă a coborât la adâncimea h=3 km, stabilind un record de scufundare. Se știe că durata scufundării a fost t=137 min. Calculați viteza medie de scufundare a balenei.*

Modalitate de rezolvare:

- faceți un tabel precum cel din figura alăturată;
- în celula B2, calculați distanța în metri (=B1*1000);
- în celula B5, calculați timpul (durata scufundării) în secunde (=B4*60);
- știind că formula vitezei este $viteza = \frac{distanța}{timp}$ (în cazul nostru, distanța este adâncimea până la care s-a scufundat balena), în celula B8 calculați viteza medie de scufundare a balenei (=B2/B5) și veți afla rezultatul final: 0.36m/s.

	A	B	C
1	Adâncimea h:	3	km
2		3000	m
3			
4	Timpul t:	137	min
5		8220	s
6			
7	V=h/t	=B2/B5	
8		0.36	m/s

2.2. Rezolvarea unei probleme de matematică

Se dă următoarea problemă de matematică: *Stabiliți care dintre numerele următoare sunt divizibile cu 3: 21, 100, 51, 14, 55, 88, 65, 77, 13, 79.*

Modalitate de rezolvare:

- faceți un tabel precum cel din figura alăturată;
- în coloana A introduceți toate numerele din problemă pe care trebuie să le testați;
- în coloana B calculați rezultatul împărțirii fiecărui număr la trei ($=A2/3$) și replicați aceeași formulă până la ultimul număr (79, din rândul 11);
- în acest moment, puteți spune care numere sunt divizibile cu trei (cele care au dat un număr întreg la împărțire) și care nu (cele care nu au dat număr întreg în urma operației).

	A	B	C
1	Numarul	Nr/3	DIVIZIBIL/NEDIVIZIBIL
2	21	7	DIVIZIBIL
3	100	33.33333	NEDIVIZIBIL
4	51	17	DIVIZIBIL
5	14	4.666667	NEDIVIZIBIL
6	55	18.33333	NEDIVIZIBIL
7	88	29.33333	NEDIVIZIBIL
8	65	21.66667	NEDIVIZIBIL
9	77	25.66667	NEDIVIZIBIL
10	13	4.333333	NEDIVIZIBIL
11	79	26.33333	NEDIVIZIBIL



Descoperiți

Pentru a întregi problema de matematică 2.2., descoperiți în cele ce urmează o altă funcție interesantă:

Funcția **INT** – ÎNTREG

Descriere: rotunjește prin lipsă la cel mai apropiat număr întreg.

Sintaxa este: **=INT(număr)**

De exemplu: **=INT(3, 44)** întoarce valoarea 3.



Aplicați și investigați

La problema 2.2. de mai sus, automatizați partea de decizie. În dreptul numerelor divizibile cu 3 scrieți **DIVIZIBIL**, în celelalte cazuri **NEDIVIZIBIL**.

Puteți obține direct în coloana C calitatea unui număr de a fi sau nu divizibil cu trei, astfel:

- introduceți în celula C2 funcția **=IF(INT(B2)=B2, "DIVIZIBIL", "NEDIVIZIBIL")**;
- replicați această funcție până în celula C11 și veți obține automat **DIVIZIBIL** sau **NEDIVIZIBIL** pentru fiecare număr cerut de problemă.



Exersați

1 Utilizând o foaie de calcul în MS Excel, calculați suma numerelor de la 1 la 10, apoi suma numerelor de la 1 la 100.



2 Faceți corespondența între funcțiile MS Excel din prima coloană și definițiile lor din a doua coloană:

- | | |
|-------------------|--|
| a. SUM | I. funcție de decizie |
| b. AVERAGE | II. funcție care întoarce valoarea cea mai mică dintr-un set de valori |
| c. MAX | III. funcție care calculează media aritmetică a unui grup de numere |
| d. MIN | IV. funcție care întoarce valoarea cea mai mare dintr-un set de valori |
| e. IF | V. funcție care adună valori |

3 Faceți un tabel în MS Excel cu toți colegii din clasa voastră, adăugând înălțimea fiecăruia în centimetri, și găsiți care este cel mai înalt, folosind funcția corespunzătoare.

4 Care este funcția pentru calculul mediei în MS Excel? Dar sintaxa ei?

5 Realizați un tabel în MS Excel cu situația școlară la matematică a tuturor elevilor din clasa voastră, calculați mediele lor și determinați media minimă, respectiv pe cea maximă.

6 Faceți un tabel în MS Excel cu un traseu de făcut prin țară cu bicicleta într-o excursie de 10 zile. Introduceți orașele de vizitat, distanțele de parcurs până la fiecare oraș și timpii estimați pentru parcurgerea fiecărei etape. Calculați și afișați: distanța totală, timpul total de mers, etapa cu distanța minimă și etapa cu distanța maximă.

Modalități grafice de utilizare a registrelor de calcul



Descoperiți

Am văzut că folosind un program de calcul tabelar, putem face simplu și rapid o multitudine de calcule. Putem chiar rezolva și probleme la diferite materii (matematică, fizică etc.).



Rețineți

O facilitate foarte importantă a programelor de calcul tabelar este aceea de a crea reprezentări grafice (numite și „diagrame”) care să ne ajute să ne facem o imagine clară și rapidă despre diferite date din tabele.



Programul MS Excel ne pune la dispoziție mai multe modele de grafice predefinite:

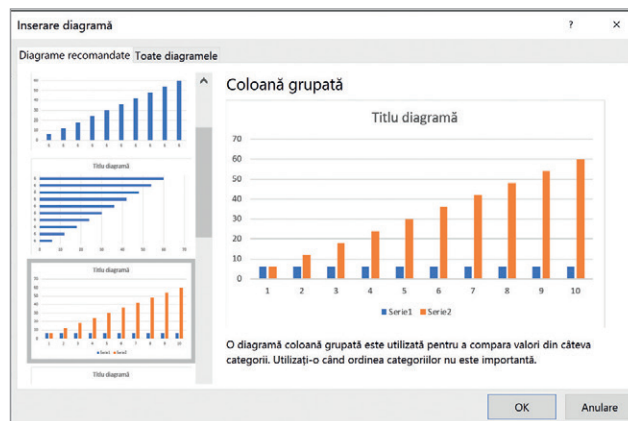
	<p>Linie 2-D</p> <p>Linie 3D</p> <p>Suprafață 2-D</p> <p>Arie 3D</p>	<p>Structură radială 2-D</p> <p>Structură radială 3D</p> <p>Structură inelară</p>
<p><i>Diagramă coloană sau bară</i></p> <p>Utilizați acest tip de diagramă pentru a compara vizual datele din câteva categorii.</p>	<p><i>Diagramă cu linii sau arie</i></p> <p>Utilizați acest tip de diagramă pentru a arăta tendințele în timp sau pe categorii.</p>	<p><i>Diagramă radială sau inelară</i></p> <p>Utilizați acest tip de diagramă pentru a reprezenta grafic proporțiile unui întreg.</p>



Sunt și alte tipuri de diagrame disponibile; toate se regăsesc în fila **Inserare**, grupul **Diagrame**. Faceți un clic pe săgeata din dreapta-jos a grupului și descoperiți-le.

Cum pot fi create grafice (diagrame) în Excel, în doar patru pași simpli:

- 1 Faceți un tabel în care puneți datele care vor fi reprezentate grafic.
- 2 Selectați tot tabelul.
- 3 Accesați fila **Inserare**, grupul **Diagrame**.
- 4 Alegeți graficul care se potrivește cel mai bine situației și care arată așa cum doriți din fereastra **Inserare diagramă** care se deschide.



Aplicați și investigați

- 1 Să luăm următorul scenariu: avem o școală cu 1 300 de elevi, facem un tabel în care trecem câte medii avem în școală din fiecare (de la 1 la 10), dorind să reprezentăm grafic distribuția mediilor.

	A	B	C	D	E	F	G	H	I	J	K
1	Media	1	2	3	4	5	6	7	8	9	10
2	Numărul mediilor	0	0	0	36	132	234	243	292	267	96



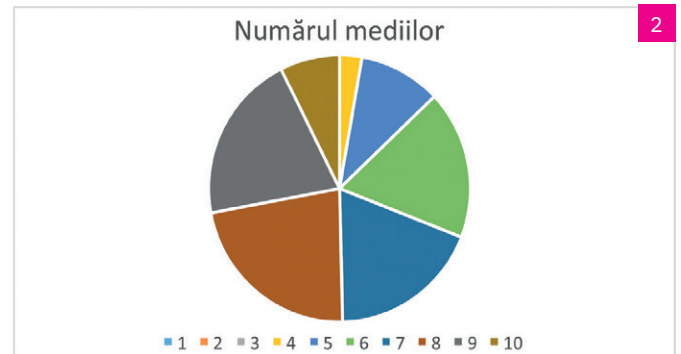
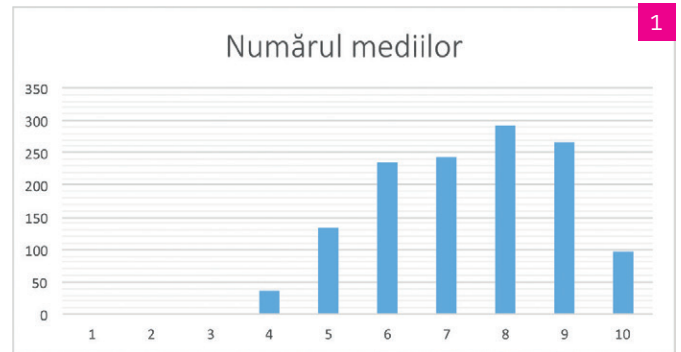
Modalitate de rezolvare:

Selectați întregul tabel, apoi mergeți în fila **Inserare** și, din grupul **Diagrame**, alegeți diagrama cu bare de la **Diagrame recomandate**. Vi se va afișa o diagramă precum cea alăturată (figura 1).

Veți obține în acest fel o diagramă din care se poate vedea cu ușurință repartizarea mediilor din școală. Puteți observa imediat că mediile de 8 sunt cele mai numeroase și mediile de 4 sunt mai rare, mult mai puține comparativ cu celelalte.

Dacă veți dori să evidențiați ponderea mediilor de 4 din totalul mediilor din școală, de exemplu, este potrivită o diagramă radială (pe care o puteți alege tot de la **Diagrame recomandate** sau direct de la **Diagrame cu structură radială**).

Observați că ponderea mediilor de 4 (reprezentate cu galben) din întreg (din totalul mediilor) este mică (figura 2).



- 2 Aveți următoarea problemă de fizică: *Dan a înregistrat într-un tabel poziția automobilului în care se afla față de casa lui, la diferite momente. Datele obținute de el se regăsesc în tabelul de mai jos.*

t(s)	0	1	2	3	4	5	6	7	8
x(m)	10	11.5	13	14.5	16	17.5	19	20.5	22

Reprezentați graficul mișcării și calculați viteza automobilului în intervalul de timp considerat.

Modalitate de rezolvare:

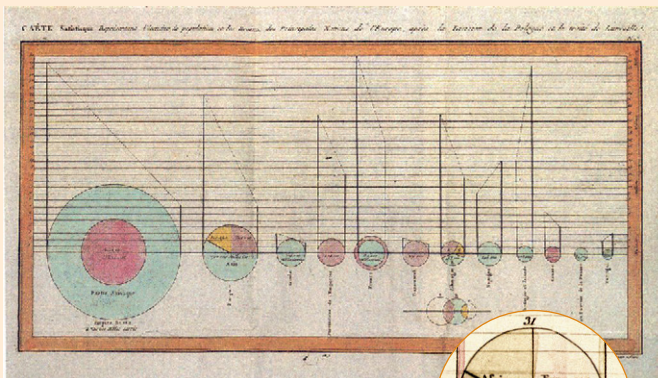
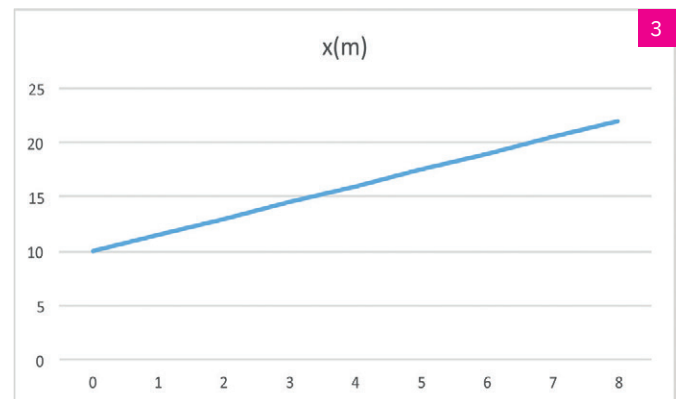
Tot ce aveți de făcut pentru reprezentarea graficului mișcării este să introduceți datele într-o foaie de calcul în MS Excel, să selectați întregul tabel și să creați graficul corespunzător tabelului (figura 3).

Pentru calcularea vitezei, trebuie să cunoașteți formula vitezei, $v=d/t$. Puteți introduce într-o celulă a foii de calcul formula:

$$=(22-10)/8.$$

Va rezulta $v=1.5$ m/s.

Puteți, de asemenea, să folosiți adresarea celulelor pentru calculul vitezei.



Primele grafice plăcintă (desen întreg și detaliu)

ȘTIAȚI CĂ...?

O diagramă radială mai poartă numele sugestiv de diagramă **plăcintă** (*pie chart*), datorită formei sale circulare și a împărțirii în „felii” pentru a ilustra proporțiile numerice.

Primele grafice de acest fel datează din anul 1801 și au fost prezentate de către William Playfair, inginer și economist politic scoțian, în lucrarea sa *Statistical Breviary*.

Una dintre diagramele sale arată proporțiile Imperiului Otoman localizat în Asia, Europa și Africa, înainte de 1789.

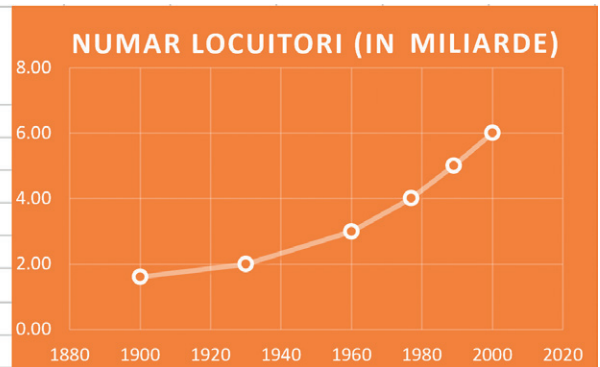
3 La geografie, trebuie să reprezentați grafic evoluția numărului de locuitori ai planetei. La o căutare rapidă pe Internet, veți găsi astfel de informații: *În ceea ce privește creșterea populației, a fost nevoie de întreaga istorie ca populația lumii să atingă 1,6 miliarde la anul 1900, totalul acesteia ajungând în 1930 la două miliarde. Al treilea miliard s-a adăugat până în anul 1960, iar al patrulea miliard până în anul 1977. În doar 12 ani, populația a ajuns la 5 miliarde, respectiv până în anul 1989, depășind 6 miliarde în anul 2000.*

Modalitate de rezolvare:

Pentru a reprezenta grafic aceste informații, faceți un tabel în care să treceți, pentru fiecare an menționat în text, numărul de locuitori ai planetei noastre.

Apoi selectați întregul tabel și inserați diagrama prin puncte, dintre cele recomandate de MS Excel. Veți obține o reprezentare grafică sugestivă pentru creșterea populației planetei, așa cum vedeți în imaginea alăturată.

Populația lumii	Numar locuitori (in miliarde)
1900	1.60
1930	2.00
1960	3.00
1977	4.00
1989	5.00
2000	6.00



Exersați

1 Cum vă găsiți lucrurile în casă? Analizați-vă cum procedați atunci când căutați ceva.

- a Căutați până găsiți singuri;
- b Renunțați;
- c MAMAAAAAAA!

Treceți într-un tabel realizat în Excel aceste situații dintr-o săptămână. Apoi reprezentați grafic rezultatul. Cum vi se pare?

Indicație: Realizați în Excel un tabel cu două coloane și patru rânduri, precum cel alăturat, completați numerele pentru fiecare caz, apoi mergeți în fila **Inserare** și, de la grupul **Diagrame**, alegeți o diagramă **Structură radială**.

Soluția aleasă	Nr de cazuri din fiecare soluție
Caut și găsesc singur	
Renunț	
MAMAAAAAAA!	

- 2 Într-un tabel cu date creat în MS Excel, pentru care ați realizat și o diagramă, modificați datele dintr-o celulă. Ce se întâmplă cu diagrama? Marcați răspunsul corect:
 - a Diagrama a fost ștearsă;
 - b Diagrama a fost modificată în funcție de noul conținut al celulei;
 - c Diagrama a fost dublată, cea nouă fiind actualizată după noul conținut al celulei modificate.

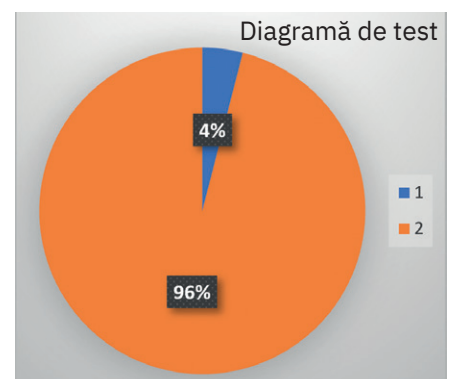
3 Realizați un tabel cu două rânduri și zece coloane în MS Excel, completați-l cu date numerice oarecare, apoi creați două diagrame (una cu bare și una radială) pentru reprezentarea grafică a datelor din tabel.

4 Care sunt cele trei tipuri de diagrame cu structură radială din MS Excel?

5 Priviți diagrama alăturată („Diagramă de test“). Cum o interpretați?

6 Răspundeți la următoarele întrebări legate de realizarea modelelor grafice (diagramelor) în MS Excel:

- a Unde se găsește grupul **Diagrame**?
- b Când este potrivit să utilizați diagrama radială?
- c Când este potrivit să utilizați diagrama cu linii?



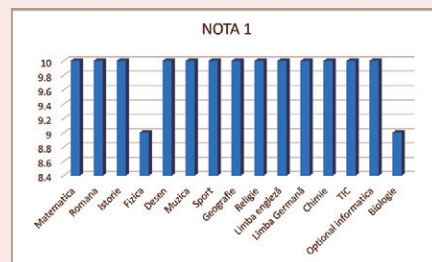
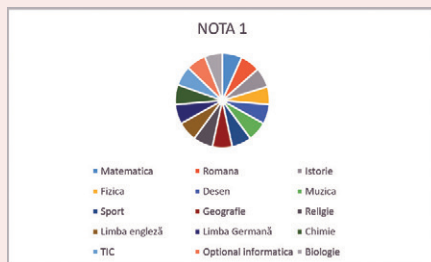
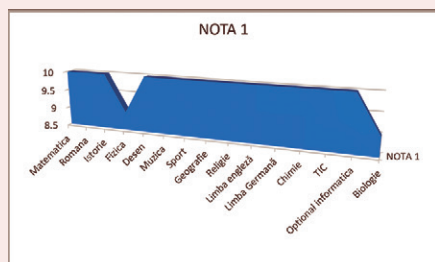
7 **Portofoliu.** Evoluția populației României.

Știm că în 1980 populația României era de aproximativ 21 000 000 locuitori. Realizați un tabel cu două coloane („Anul“ și „Populația României“). În prima coloană completați o serie de ani, începând de la 1980, din 3 în 3, până la 2037. În a doua coloană completați numărul de locuitori ai României, pornind de la populația existentă în 1980, cu un procent de scădere de 10% la fiecare 3 ani (folosiți formula de calcul potrivită). Realizați apoi diagrama care reprezintă grafic evoluția populației României în această perioadă. Cum ar arăta diagrama dacă procentul de scădere ar fi de 3%?

Recapitulare

Rezolvați în caiete sau pe calculator.

- Ce este MS Excel?
 - un limbaj de programare;
 - un program de calcul tabelar;
 - un program de editat texte.
- Numiți și descrieți trei dintre filele aplicației MS Excel.
- În MS Excel, adresa C3 se referă la:
 - primele 3 celule din coloana C;
 - celula aflată pe linia a treia, coloana C;
 - pagina C3 din registrul de calcul.
- Coloanele în MS Excel se numerotează cu:
 - cifre romane;
 - litere mari;
 - cifre arabe.
- Pentru introducerea formulelor în celule în MS Excel, trebuie să:
 - începem formula cu semnul „=”;
 - scriem formula, urmată de semnul „=”;
 - scriem direct formula de calcul.
- O celulă dintr-o foaie de calcul a MS Excel poate conține:
 - texte;
 - formule;
 - numere.
- Funcția SUM din MS Excel se folosește la:
 - realizarea sumei tuturor celulelor din registrul de calcul;
 - realizarea sumei celulelor furnizate ca argument al funcției;
 - realizarea sumei celulelor care conțin valori exprimate în Euro.
- Extensia unui fișier creat cu MS Excel este:
 - .exe;
 - .xlsx;
 - .doc.
- Care dintre următoarele funcții calculează media aritmetică a celulelor C1, C2 și C3 în MS Excel?
 - =AVERAGE(C1:C3);
 - =AVERAGE(C1, C4);
 - =SUM(C1, B3)/3.
- Realizați un tabel în MS Excel cu numele a 10 fluvii din lume și cu lungimile lor, apoi ordonați-le descrescător, în funcție de lungime.
- Realizați în MS Excel un tabel cu inventarul sertarului vostru cu rechizite (creioane, pixuri, carioci, caiete etc.) și calculați totalul obiectelor, folosind funcția corespunzătoare.
- Faceți corespondența corectă între diagrame și numele lor:
 -
 -
 -



1. Diagramă coloană 3D (utilizată pentru a compara valori din mai multe categorii)

2. Diagramă arie 3D (utilizată pentru a reprezenta tendințele pe categorii)

3. Diagramă cu structură radială (utilizată pentru a reprezenta proporțiile unui întreg)

Autoevaluare

Rezolvați în caiete sau pe calculator.

- 1 Ce este un program de calcul tabelar?
- 2 În care dintre următoarele tipuri de date poate fi formatată o celulă dintr-o foaie de calcul în MS Excel?
 - a text;
 - b număr;
 - c monedă;
 - d dată;
 - e procent;
 - f fracție.

- 3 Realizați, în caiete, corespondența dintre noțiunile date și descrierile lor, din tabelul de mai jos:

a. Bara de titlu	1. include toate comenzile aplicației MS Excel grupate pe file, pe tipuri de acțiuni
b. Panglica de butoane	2. afișează numele celulei în care suntem poziționați
c. Caseta de nume	3. foaia de calcul, zona în care introducem date, formule, realizăm diagrame
d. Zona de lucru	4. conține butoanele de minimizare, restaurare, închidere, salvare, anulare acțiune, refacere acțiune și titlul registrului MS Excel

- 4 Alegeți, din lista următoare, elementele principale ale interfeței utilizator a MS Excel:
 - a Bara de defilare verticală;
 - b Panglica;
 - c Zona de lucru;
 - d Bara de desenare;
 - e Instrument de panoramare;
 - f Eticheta foi de calcul.
- 5 Dintre aplicațiile următoare, transcrieți-le în caiete pe cele care sunt programe de calcul tabelar:
 - a MS Excel;
 - b MS Outlook;
 - c Libre Office Calc;
 - d Excel Online;
 - e Ashampoo PlanMaker;
 - f MS Word.
- 6 Numiți două avantaje practice ale stocării unui registru de calcul în cloud.
- 7 Scrieți **DA** sau **NU** pentru fiecare dintre afirmațiile de mai jos:
 - a Textul dintr-o celulă poate fi: drept (simplu), aplecat, îngroșat, subliniat. DA / NU
 - b Butonul **Sortare** din fila **Aspect pagină** sortează datele din tabel după criteriile selectate. DA / NU
 - c Funcția **SUM** adună valori. Se pot aduna valori individuale sau referințe de celule. DA / NU
- 8 Completați, în caiete, enunțurile următoare, folosind cuvintele potrivite:
 - a Fila **Inserare** conține necesare pentru adăugarea de alte elemente de conținut într-o foaie de, spre exemplu,, ilustrații,, hărți, video, legături către diferite pagini web,, simboluri etc.
 - b Pentru a tipări registrul de, trebuie să alegeți comanda, aflată în fila **Fișier**. Documentul se poate tipări la o instalată în prealabil.
 - c Funcția **IF** este funcția de, Este una dintre cele mai populare funcții din și vă permite să efectuați comparații între o valoare și valoarea la care vă așteptați. Așadar, o funcție **IF** poate avea rezultate.
- 9 Dați trei exemple de operații elementare din MS Excel pentru lucrul cu un registru de calcul.
- 10 Ce sunt și la ce folosesc diagramele din MS Excel?

Punctaje:

1 – 10 puncte;
2 – 10 puncte;
3 – 10 puncte;
4 – 10 puncte;
5 – 10 puncte;

6 – 10 puncte;
7 – 10 puncte;
8 – 10 puncte;
9 – 10 puncte;
10 – 10 puncte.

Punctaj maxim: 100 puncte.
Timp de lucru: 30 minute.

Răspunsurile se găsesc
la finalul manualului.

Insuficient: 0-50 p
Suficient: 51-60 p
Bine: 61-80 p
Foarte bine: 81-90 p
Excelent: 91-100 p

U2

Aplicații de realizare a paginilor web

L1 Pagini web – exemple, analiză, structură

L2 Aplicații specializate pentru crearea paginilor web – elemente de interfață și facilități

L3 Crearea unei pagini web. Publicarea unui site de o pagină

L4 Construirea unui site Internet cu mai multe pagini. Lucrul colaborativ

Recapitulare

Autoevaluare



WWW

Pagini web – exemple, analiză, structură



Amintiți-vă

Internetul este sistemul global de rețele de calculatoare interconectate care utilizează un anumit protocol de comunicare (TCP/IP) pentru a conecta dispozitivele din întreaga lume. Este o rețea de rețele.

O **pagină web** este o resursă din spațiul web (WWW) din Internet, de obicei în format HTML și având hyperlinkuri (hiperlegături) pentru navigarea simplă (cu un singur clic de mouse) de la o pagină sau secțiune de pagină la alta. Pagina web se numește astfel („pagină”), deoarece se aseamănă într-o oarecare măsură cu o pagină de ziar.

Un **site web** este o grupare pe o anumită temă a mai multor pagini web care sunt conectate între ele prin hyperlinkuri. Există site-uri web care conțin o singură pagină web, acestea fiind cazuri particulare.



Descoperiți

Cu toții „navigăm” pe Internet, sărim cu ușurință de la o pagină web la alta, de la un site la altul, în căutarea informațiilor dorite. Site-uri cu informații, enciclopedii, site-uri de sport, site-uri de știri, bloguri, site-uri cu jocuri, cu filme, site-uri de modă, magazine online sau orice altceva, toate ne sunt foarte la îndemână.

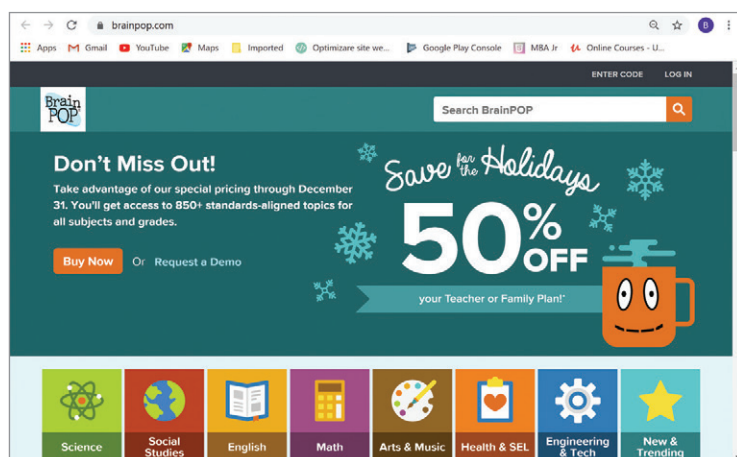
Dați cinci exemple de site-uri pe care le vizitați de obicei. Ce tip de informații găsiți aici?

Se nasc însă mai multe întrebări. Cum și cu ce programe anume se face un site Internet? Care sunt elementele specifice unei pagini web? La toate aceste întrebări și la multe altele legate de subiectul „site-uri web” vs. „pagini web” vom găsi răspunsurile în paginile următoare.

Să vedem împreună care sunt elementele de structură ale unei pagini web. Cel mai bine ar fi să analizăm câteva exemple, să vizităm pagini web aparținând unor site-uri diferite.

Deschideți și analizați cu atenție următoarele site-uri (dar puteți deschide orice alte pagini, pe care voi le utilizați de obicei sau unele noi, pe care le deschideți acum pentru prima dată). Alegeți două dintre acestea și notați ce elemente sunt comune în alcătuirea lor (ce se află în partea de sus, ce se află în partea de jos):

- www.manuale.edu.ro – site-ul unde găsim toate manualele noastre digitale
- www.aboutyou.ro – un site de modă, atât pentru fete, cât și pentru băieți
- www.spotify.com – una din sursele noastre de muzică – putem găsi și asculta muzica favorită
- www.scratch.mit.edu – site-ul mediului de programare Scratch
- www.brainly.ro – site-ul de ajutor la teme, în care o comunitate mare de elevi, părinți și profesori își împărtășesc cunoștințele
- www.math-pdr.com – un site cu o colecție de lecții de matematică (în format video, în limba română), de ajutor în aprofundarea cunoștințelor de matematică pentru clasele I – XII
- www.brainpop.com – un site pe care putem învăța despre evenimente istorice, știință și chiar economie, cu ajutorul unui roboțel simpatic și al prietenilor săi care ne prezintă totul sub formă de animații și dialoguri captivante



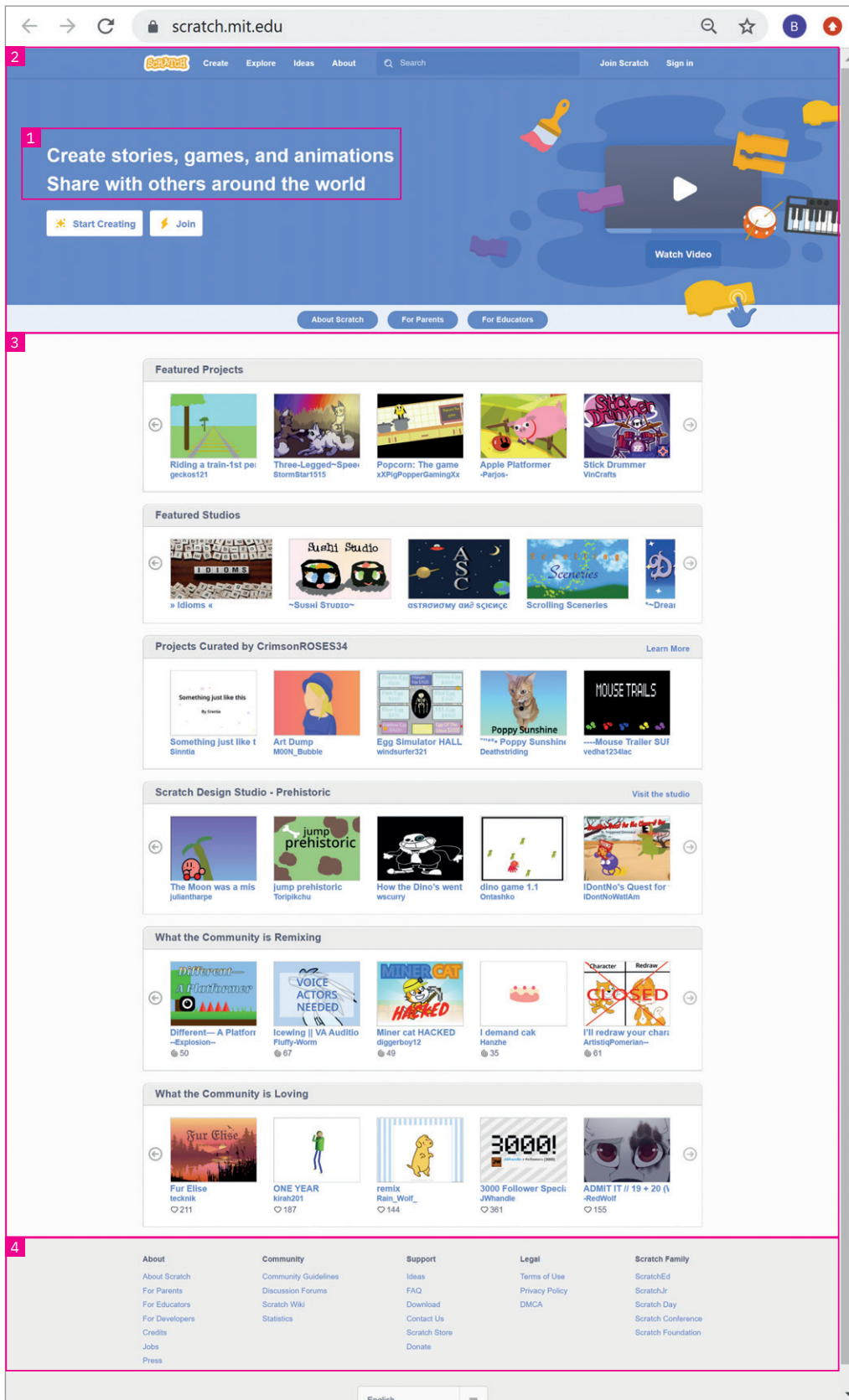
Exemplificare site www.brainpop.com



Vizitând paginile web ale site-urilor de mai sus, observăm că, în general, o pagină web este formată din câteva elemente de structură principale:

- 1 – titlul paginii;
- 2 – antetul paginii;
- 3 – corpul paginii;
- 4 – subsolul paginii.

Să luăm ca exemplu prima pagină a site-ului Scratch.



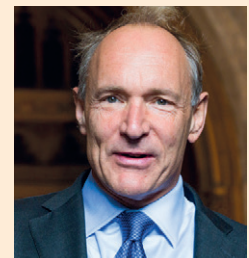
Exemplificare structură site pagină web (site Scratch)

Lucrați în perechi. Observați zonele mai sus menționate, clar delimitate pe această pagină. Comparați antetul, corpul și subsolul paginii web din punctul de vedere al dimensiunii acestora. Ce elemente sunt prezente în antet și, respectiv, în subsolul paginii exemplificate?

Observați titlul acestei pagini și poziționarea sa. În jurul lui este lăsată o zonă liberă tocmai pentru a ieși în evidență ușor.

ȘTIAȚI CĂ...?

- Viziunea lui Tim Berners-Lee (inventatorul World Wide Web) pentru ceea ce urma să devină web, înaintată spre acceptare șefului său direct, Mike Sendall, în martie 1989, într-un document numit „Managementul informației: Propunere“, nu a fost acceptată imediat. Acesta nota pe prima pagină a documentului: „Vag, dar interesant.“



Tim Berners-Lee

- Există un site care arată cât de „bătrân“ este Internetul. Puteți accesa www.howoldistheinter.net și veți vedea numărul de zile trecute de la lansarea acestuia în forma sa actuală până în prezent.
- Primul site web din lume a fost realizat în cadrul CERN (European Organization for Nuclear Research/Organizația Europeană de Cercetare Nucleară) de către Tim Berners-Lee și a fost lansat la 6 august 1991. El era dedicat proiectului WWW (World Wide Web) și rula pe calculatorul lui Berners-Lee, NeXT. Acest site poate fi accesat și astăzi la adresa: <http://info.cern.ch/hypertext/WWW/TheProject.html>.

✓ Rețineți

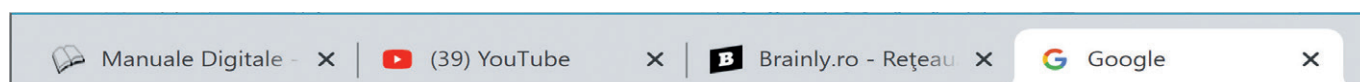
În general, în **antet** găsim **titlul paginii** și elemente de identitate vizuală a paginii.

În **corpul paginii** regăsim informația principală. Această zonă este cea mai importantă parte a paginii web, ea ocupând cel mai mult spațiu din pagină, oferind informație consistentă.

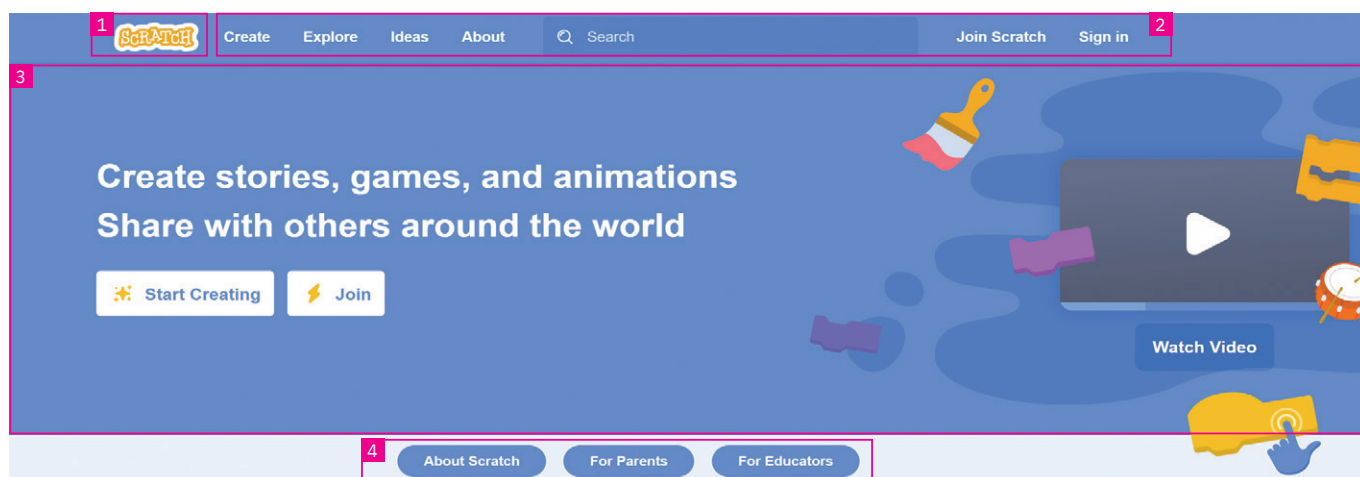
Subsolul este ultima parte a paginii. El poate să conțină câteva informații sumare: un meniu secundar, date de contact și linkuri către platforme sociale.

O pagină web poate exista și poate funcționa fără oricare din aceste trei elemente (antet, corp, subsol), dar va fi ca o compunere căreia îi lipsește introducerea, cuprinsul sau încheierea!

Se mai poate identifica un element interesant, numit **tag**, care anunță conținutul unei pagini web deschise și apare alături de o iconiță sugestivă, în fila browserului corespunzătoare paginii deschise a site-ului web.



Dacă analizăm în amănunt, în zona antetului mai observăm câteva elemente care se regăsesc la majoritatea site-urilor, respectându-se un fel de șablon:



- 1 – logo;
- 2 – meniuri;
- 3 – zona imaginii sau imaginilor de antet (după cum observați, în această zonă putem regăsi și unele butoane cu diferite funcții);
- 4 – al doilea rând de meniuri (poate lipsi la unele pagini web).

La unele pagini web, în antet veți mai găsi și un instrument de căutare. Introduceți aici textul de căutat pentru a identifica mai ușor informația dorită în textul paginii.


Tot în antet, unele site-uri alocă un spațiu mai mare unui element grafic folosit pentru prezentare, numit **slider** (sau *carusel de imagini*). În terminologia utilizată în web design, termenul „slider” este folosit pentru o prezentare de imagini sugestive introdusă în pagina web. Este vorba de un set de imagini care se schimbă automat sau nu, cu diferite efecte de trecere între ele, menite să atragă atenția, să încânte și să convingă utilizatorul să parcurgă conținutul paginii. Există o mulțime de unelte (software) ajutătoare cu care puteți crea astfel de slidere pe care să le introduceți în paginile voastre web.

Exemplu de slider

Unele site-uri păstrează fixă zona de antet pe toate paginile lor, altele o au doar pe prima pagină (pagina **Acasă**), iar pe restul paginilor o au într-o formă simplificată, care ocupă mai puțin loc, în general fără zona de imagine sau de *carusel de imagini* din antet.



Un avantaj clar al utilizării unui slider este faptul că puteți plasa într-o manieră vizuală informațiile pe care doriți să le scoateți în evidență pentru a fi recepționate de către vizitatorii paginii simplu și rapid.

 Un slider poate rula imagini în mod automat fără vreo acțiune din partea utilizatorului, schimbându-le la un interval de timp predefinit, cu anumite efecte de trecere între ele. Sliderele pot și să răspundă la interacțiunea utilizatorului, cum ar fi clic sau glisare pentru a vizualiza imaginile următoare sau anterioare. În plus, ele pot avea și butoane sau miniaturi pe care utilizatorii pot face clic pentru a vizualiza o anumită imagine din slider.

Acum aveți mult mai multe cunoștințe despre elementele paginilor web, cunoștințe care vă vor face să priviți paginile web cu alți ochi, să fiți analiști fini ai paginilor pe care le veți vizita.

IMPORTANT! Trebuie să aveți grijă cum vă protejați împotriva fraudelor informatice atunci când deschideți diferite pagini web. Securitatea cibernetică este vitală.

Mare atenție: trebuie să acordați o importanță deosebită informațiilor voastre personale! Nu vă lăsați păcăliți de acele pagini web care încearcă prin diferite formulări destul de convingătoare (la o primă vedere) să obțină de la voi astfel de informații. Sub termenul general de **inginerie socială** înțelegem, în legătură cu securitatea informațiilor, manipularea psihologică a oamenilor pentru a-i face să întreprindă acțiuni prin care să își divulge către persoane rău intenționate diferite informații confidențiale (începând cu adrese de e-mail sau numere de telefon, adresa de domiciliu sau codul numeric personal și mergând până la parole sau informații legate de conturi bancare). În concluzie, este o formă de escrocherie prin care se urmărește colectarea de informații care ulterior pot fi folosite de către cei care le-au colectat, cauzând diferite și însemnate pagube celor care au căzut în capcană.



Rețineți

Reguli pentru o navigare mai sigură printre paginile de net:

- nu accesați pagini web care par nesigure încă de la prima vedere;
- verificați secțiunea de contact a site-urilor web (adresă, număr de telefon, e-mail);
- verificați destinația reală a linkurilor prin trecerea cursorului mouse-ului peste acestea și vizualizarea adresei reale în partea stângă-jos a browserului;
- nu apăsați pe linkurile din cadrul ferestrelor de tip pop-up (fereastră nouă de browser care afișează în general reclame);
- verificați existența „https://” în partea de început a adresei web, înainte de a introduce informații personale;
- nu instalați programe decât din proprie inițiativă, nu la propunerea altcuiva! Multe amenințări online vin sub forma cererilor de a da clic pe un anumit link sau de a deschide atașamentul unui mesaj e-mail. Sunt pagini web ce deschid ferestre pop-up care vă cer să rulați un scanner de securitate sau să instalați un player pentru a putea vizualiza diverse conținuturi (care pot părea atrăgătoare; escrocii știu să vă stârnească interesul). Evitați să dați curs unor asemenea cereri;
- dacă este necesar să instalați un program, încercați să îl descărcați direct de la sursă (de pe site-ul web al producătorului) și nu de pe site-uri web terțe.



Aplicați și investigați

- 1 Deschideți pagina www.pbinfo.ro. Identificați antetul paginii. Antetul este cel pe care îl aveți colorat cu albastru în partea de sus. Identificați în subsolul paginii butonul de contact. Subsolul acestei pagini este colorat în gri. Identificați corpul paginii. Corpul paginii este reprezentat de tot ce este cuprins între antet și subsol. Găsiți în corpul paginii cele două câmpuri care se completează pentru autentificare. Care sunt acestea?
- 2 Alegeți o pagină web preferată și identificați-i elementele componente.



Exersați



- 1 Scrieți, în caiete, A (Adevărat) sau F (Fals), în funcție de valoarea de adevăr a propozițiilor:

a O pagină web fără subsol nu poate funcționa.	A / F
b Este obligatoriu ca în antetul unei pagini web să existe instrumentul de căutare de text.	A / F
c Titlul unei pagini web anunță conținutul respectivei pagini.	A / F
- 2 Dați trei exemple de bune practici pentru o navigare mai sigură printre paginile web din Internet.
- 3 **Portofoliu.** Realizați o prezentare PowerPoint referitoare la momente din istoria paginilor web. Evidențiați schimbările de structură suferite în timp. Prezentarea trebuie să conțină imagini și sunete, eventual înregistrate de voi.

Aplicații specializate pentru crearea paginilor web – elemente de interfață și facilități



Descoperiți

Am văzut în lecția anterioară elementele de structură ale unei pagini web. Dar cum și cine poate crea o pagină web? Oare noi putem? Răspunsul este DA! Să vedem cum.

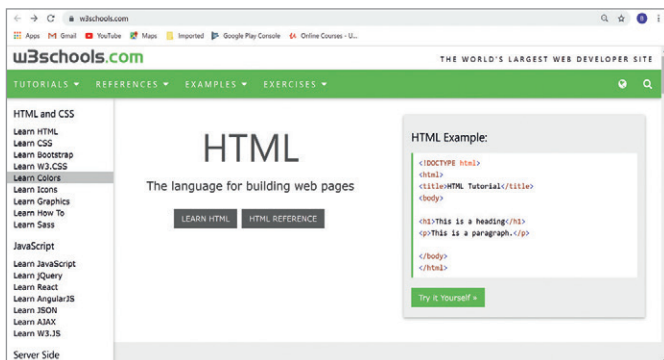


Rețineți

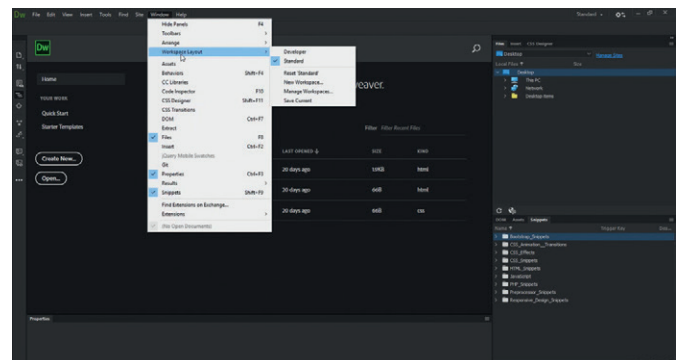
Există două posibilități total diferite și clar definite pentru crearea unei pagini web, dar și combinații ale acestor două posibilități, diferite variante intermediare:

- 1 prima posibilitate este prin programare, prin scriere de linii de cod (de exemplu, în limbaj HTML) pentru a obține pagina web dorită;
- 2 a doua metodă are în vedere folosirea de aplicații (programe) specializate în crearea și editarea de pagini web. În cele ce urmează, vom utiliza această a doua metodă.

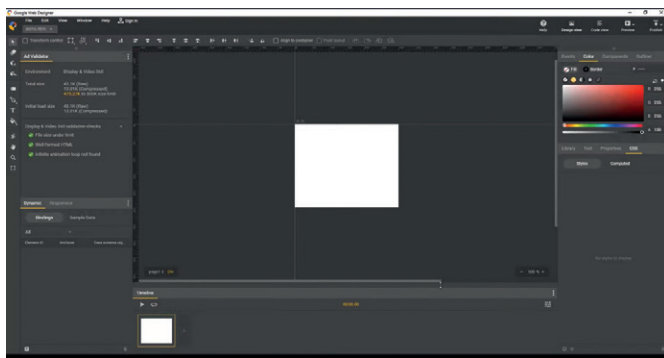
Există o multitudine de programe pentru creare și editare de pagini web. Unele se instalează pe calculatoarele voastre, altele se accesează online, în browser. Dintre cele mai utilizate și apreciate în acest moment sunt: WordPress, Wix, Kompozer, Google Sites, Dreamweaver, Blogger, Google Web Designer, Site123, Squarespace, SiteBuilder, Jimdo și multe, multe altele. W3Schools este un site foarte important pentru învățarea realizării paginilor web.



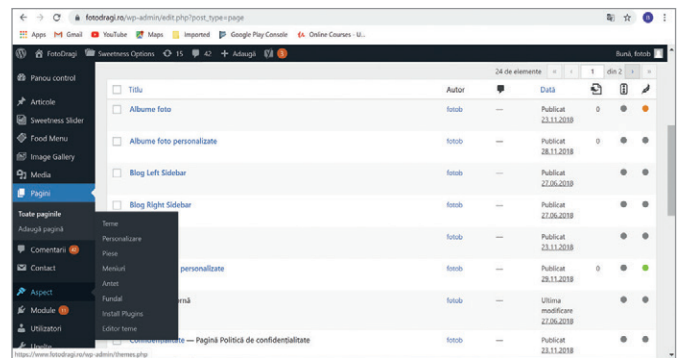
W3Schools



Dreamweaver



Google Web Designer



WordPress



Analiza unei pagini web

Chiar și din Microsoft Word puteți crea o pagină web simplă: scrieți conținutul-text pe care îl doriți într-un document nou, inserați imaginile potrivite, aranjați totul după bunul-plac, iar la final, la salvare, alegeți la tipul de fișier în care să salvați „web Page“ și veți obține un fișier html, adică o pagină web. Doar că pentru a arăta ca o pagină web modernă, actuală și pentru a avea și funcționalități similare, e nevoie de mai multă muncă. Unele funcționalități vor rămâne însă imposibil de implementat din Word.

Dintre toate aceste aplicații, noi vom analiza mai amănunțit și vom folosi în lecțiile următoare programul **Google Sites**. Îl puteți deschide accesând linkul următor: <https://sites.google.com>. Atenție, conform specificațiilor producătorului, Google Sites funcționează bine cu unul dintre următoarele browsere de Internet: Chrome, Firefox sau Safari.

Deschideți mai întâi, în editorul Google Sites, o pagină web existentă și identificați elementele din care este formată. Accesați, de exemplu, pagina <https://sites.google.com/view/site-ulmeurapid>.

Observăm *numele site-ului* („Site-ul meu rapid”) și *antetul*. În antet putem identifica titlul paginii („CUM SA FACI UN SITE IN 5 MINUTE”) și imaginea de antet (figura 1).

▶ După antet începe *corpul paginii*, conținutul paginii, partea consistentă, plină de informații. Observăm o zonă de text, cu câteva paragrafe și cu titlul aferent (figura 2).

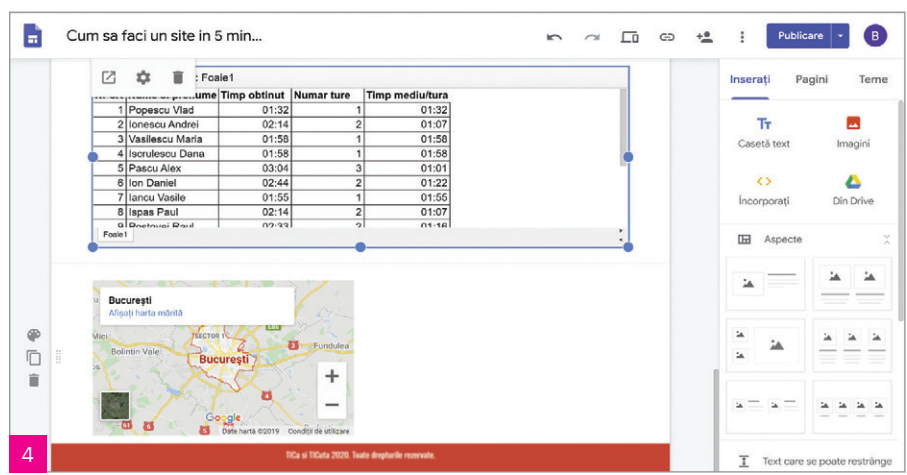
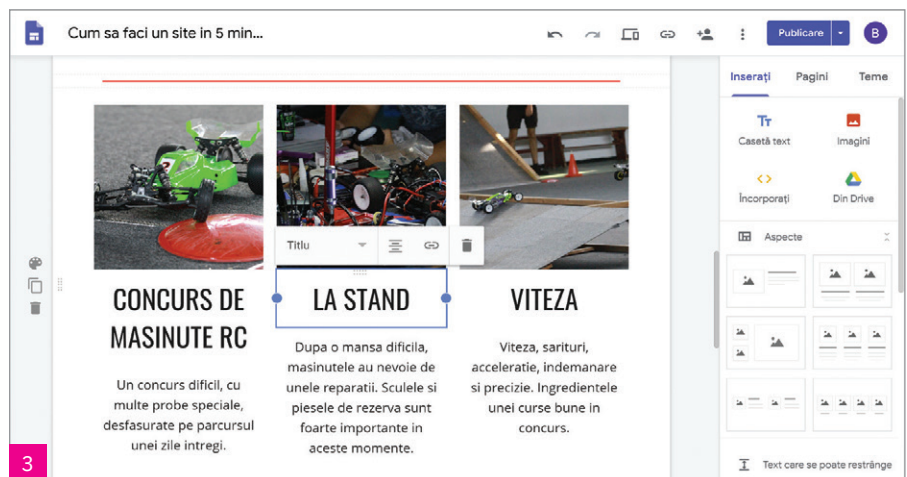
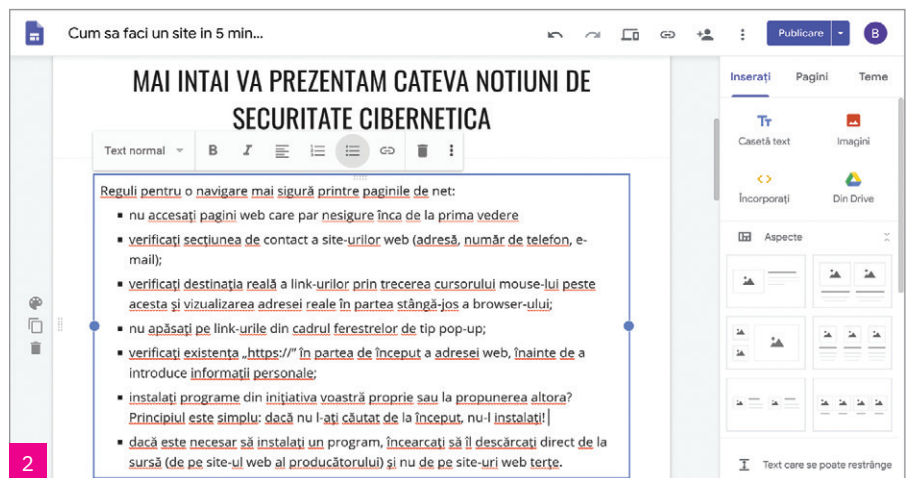
Urmează un element grafic, o linie roșie. Este numit „element separator” și are rolul (după cum îi spune și numele) de a separa vizual două zone din corpul paginii.

După separator, observăm un element al corpului paginii care seamănă cu un tabel cu două rânduri și trei coloane. Pe primul rând vedem trei imagini (sunt fotografiile de la un concurs de mașinuțe RC pilotate de colegi de-ai voștri), iar pe al doilea rând, câte un text pentru fiecare imagine, împreună cu titlul aferent (figura 3).

Această parte a paginii web este o secțiune a sa. O pagină web poate avea mai multe secțiuni.

+ Urmează un element de altă natură: un tabel. Este un tabel făcut cu un alt program, fiind inserat ulterior aici, în pagina web. Avem posibilitatea (oferită de editorul Google Sites) să inserăm diferite elemente, cum ar fi: documente (realizate într-un editor de text), filmulețe de pe YouTube, prezentări, formulare, diagrame etc. În acest caz, a fost inserat în pagina web un tabel făcut într-un program specializat în realizarea de foi de calcul. Am învățat și noi să facem astfel de tabele la capitolul dedicat foilor de calcul tabelar.

După tabel, observăm un alt element care vouă vă este familiar din diferite site-uri vizitate sau aplicații utilizate: „harta”. Se poate insera un astfel de element pentru a indica o locație importantă pentru informațiile din corpul paginii (figura 4).



Iar la final regăsim *subsolul* paginii, pe fond roșu – în cazul paginii noastre, în armonie cu dunga roșie de sub numele din antet și cu elementul separator.

Elemente de interfață Google Sites

Să discutăm acum despre editorul pe care îl vom folosi pentru realizarea paginii noastre web și elementele de interfață ale acestuia.

Ați făcut deja cunoștință cu **Google Sites**, una dintre cele mai populare aplicații de editare a paginilor web. Să analizăm împreună principalele elemente de interfață ale editorului de pagini web Google Sites.

Odată ce accesăm linkul <https://sites.google.com>, ne facem un cont (dacă nu avem deja unul) și ne autentificăm; ni se va deschide un ecran pentru gestionarea proiectelor de site-uri similar cu cel alăturat.

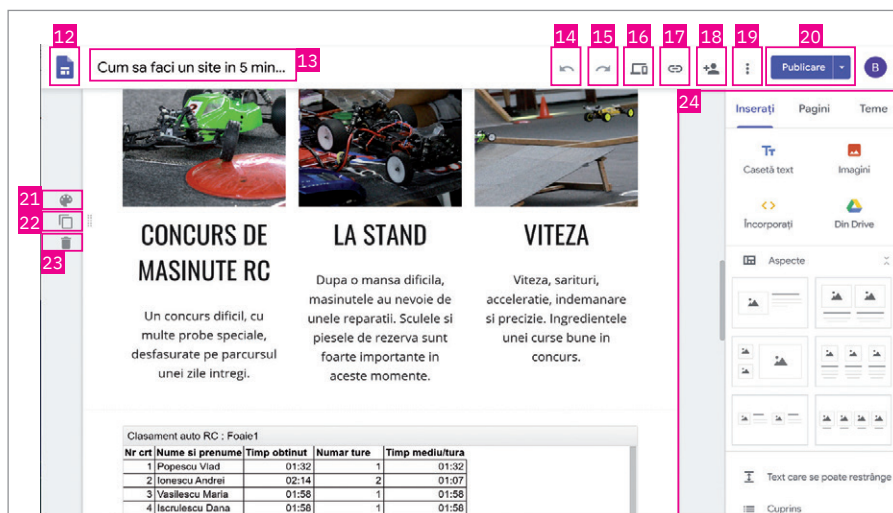
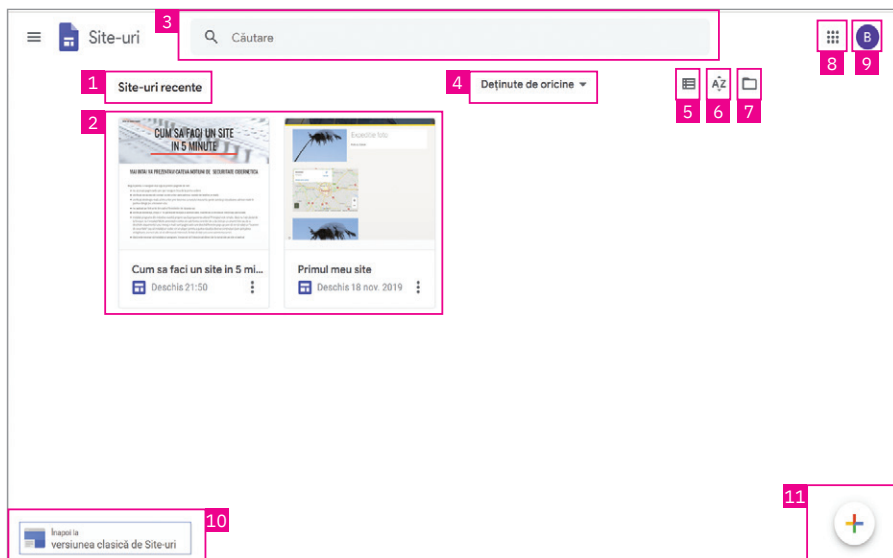
Iată ce conține:

- 1 – sub denumirea **Site-uri recente** vom regăsi ultimele proiecte la care am lucrat (primul este cel mai recent);
- 2 – **iconițele corespunzătoare proiectelor** (un clic pe iconiță deschide proiectul corespunzător);
- 3 – **bara de căutare** – ne va fi utilă atunci când vom avea multe site-uri create; doar introducem câteva cuvinte din numele proiectului căutat și ne apar toate proiectele care au în nume ceea ce s-a introdus;
- 4 – de la acest meniu putem alege site-urile **Deținute de mine** (adică site-urile proprii), **Nedeținute de mine** (adică site-urile create de alții, dar la care ni s-a dat acces, mai precis cele ce au fost create de altcineva și au fost partajate și cu noi pentru lucrul colaborativ) sau **Deținute de oricine** (adică ambele categorii de site-uri);
- 5 – butonul pentru **Schimbarea modului de afișare a iconițelor proiectelor** (**Vizualizare grilă**, cum sunt afișate în ecranul nostru, sau **Vizualizare listă**);
- 6 – butonul **Opțiuni de sortare**, de la care se pot ordona proiectele existente după diferite criterii (faceți un clic pe buton și veți descoperi care sunt opțiunile de sortare disponibile);
- 7 – butonul **Deschideți selectorul de fișiere**, un instrument care ne ajută să navigăm printre folderele și fișierele pe care le avem la dispoziție și să selectăm fișierul dorit;
- 8 – butonul **Aplicații Google** este unul general al interfeței Google, de la care putem accesa orice aplicație Google (YouTube, Hărți, Gmail, Drive, Calendar etc.);
- 9 – butonul **Contul Google**, de la care putem gestiona contul Google;
- 10 – butonul **Înapoi la versiunea clasică de Site-uri** ne schimbă interfața de lucru la versiunea anterioară, clasică a Google Sites;
- 11 – butonul + este cel de la care putem crea un nou proiect, un nou site.

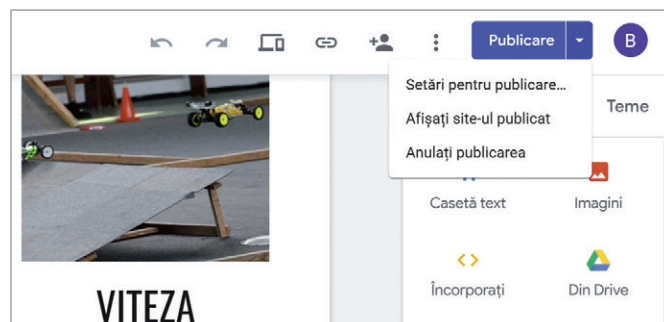
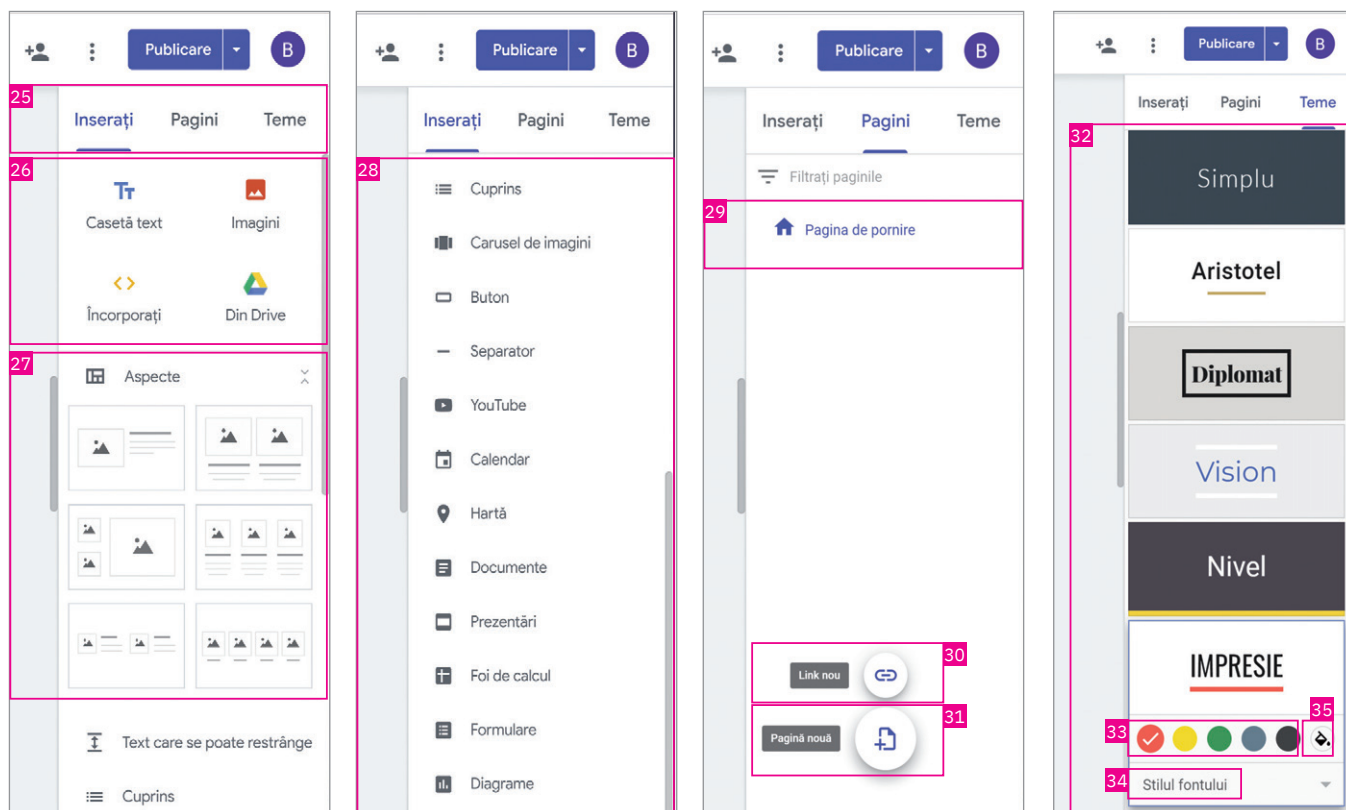
Observați că avem două proiecte (două site-uri): „Cum sa faci un site in 5 minute” și „Primul meu site”. Să deschidem proiectul „Cum sa faci un site in 5 minute”.

Elementele de interfață, care ne sunt folositoare în realizarea unui proiect, sunt cele din imaginea alăturată.

- 12 – **Pagina de pornire site-uri** este butonul de întoarcere în ecranul anterior, de unde ne putem alege un proiect sau putem crea unul nou;
- 13 – **titlul proiectului**;
- 14 – butonul **Anulați ultima acțiune**, clasicul buton **Undo** din orice editor;



- 15 – buton **Refaceți ultima acțiune** este clasicul buton **Redo** din orice editor;
- 16 – **Vizualizare** – buton care ne afișează site-ul în mod fullscreen (pe tot ecranul), așa cum arată el pe Internet în acel moment al editării;
- 17 – **Copiați linkul către site-ul publicat** – copiază (în memorie) linkul către site, iar cu ajutorul opțiunii **Lipește** putem să transmitem linkul site-ului către oricine dorim;
- 18 – **Trimiteți altor persoane** este un buton prin care putem invita prieteni sau colegi să lucreze împreună cu noi la acest proiect – le dăm acces la proiectul nostru (lucru colaborativ);
- 19 – **Mai multe** – diverse alte meniuri pe care le avem la dispoziție, de la **Statistici privind site-ul** sau **Duplicați site-ul** până la **Ajutor**.
- 20 – butonul **Publicare**, necesar pentru publicarea proiectului, cu trei opțiuni pe care le putem vedea în detaliu în imaginea de alături;
- 21 – **Fundalul secțiunii** – buton de la care se alege fundalul secțiunii de site selectate în acel moment;
- 22 – **Duplicați secțiunea** – buton pentru duplicarea secțiunii selectate, util când vrem să punem într-un site mai multe secțiuni asemănătoare;
- 23 – **Ștergeți secțiunea** – buton pentru ștergerea secțiunii de site selectate;
- 24 – în partea dreaptă a ecranului avem **Bara de unelte** (un meniu de editare cu trei file/taburi), în care se regăsește tot ce este necesar pentru realizarea site-ului;
- 25 – zona de selectare a unuia dintre cele trei taburi (**Inserați**, **Pagini** și **Teme**);
- 26 – zona de unelte de inserat casete de text, imagini, elemente de pe Internet sau elemente din Drive-ul Google precum documente text, filmulețe etc. (toate acestea se pot insera rapid și ușor în site);
- 27 – zona **Aspecte**, pentru diferite șabloane de așezare a imaginilor și a textelor.

Tabul *Inserați*Tabul *Pagini*Tabul *Teme*

În tabul **Inserați** regăsim mai multe opțiuni de adăugare a elementelor de conținut:

- 28 – zona din bara de unelte din care putem insera diferite elemente în pagina noastră: carusel de imagini (mai multe imagini care se succed), butoane, separatoare, filme de pe YouTube, calendare, hărți, documente, prezentări, foi de calcul, formulare, diagrame etc.

Tabul **Pagini** conține:

29 – zona în care putem vedea și gestiona toate paginile site-ului nostru (momentan avem doar o pagină, **Pagina de pornire** sau **Acasă**);

30 – buton de **adăugare a unei noi legături (Link nou)**; aceasta poate fi editată în orice moment pentru a realiza o legătură către o altă pagină;

31 – buton de **adăugare Pagină nouă** în site-ul nostru (într-un site construit cu Google Sites se pot adăuga un număr nelimitat de pagini).

Tabul **Teme** conține:

32 – mai multe teme predefinite dintre care se poate alege. O **temă** este un șablon pentru cum va arăta site-ul din punct de vedere coloristic, al stilului fonturilor sau stilul în ansamblu. Cu un singur clic putem schimba radical felul în care arată o pagină de site deja creată;

33 – culori pentru a putea personaliza fiecare temă;

34 – stiluri de fonturi din care putem alege ceea ce ni se pare mai potrivit pentru textul nostru;

35 – paletă de culori integrată, necesară atunci când dorim o altă culoare decât cele oferite implicit în tema aleasă. Putem chiar specifica o anumită culoare prin codul ei Hexa, o succesiune de trei perechi de numere în sistem hexazecimal precedate de # (de exemplu, portocaliu are codul #FFA500).



Aplicați și investigați

- Căutați pe Internet informații despre trei aplicații utilizate pentru editarea paginilor web. Analizați comparativ cele trei aplicații, evidențiind câte două puncte tari și câte două puncte slabe ale fiecăreia.
- Luăți o pagină web existentă, creată pentru exercițiu de profesorul vostru, și modificați-i conținutul. Încercați să modificați și să adăugați texte, să înlocuiți imagini, să adăugați diferite elemente etc. Pentru fiecare modificare, vizualizați rezultatul în programul de navigare pe Internet, pentru a-l vedea în forma în care îl vor vizualiza vizitatorii paginii respective.



Exersați

- Dați trei exemple de aplicații specializate în crearea paginilor web.



- Scrieți, în caiete, A (Adevărat) sau F (Fals), în funcție de valoarea de adevăr a propozițiilor:

a Se poate crea o pagină web folosind programul Microsoft Word.

A / F

b Se poate încărca o imagine ca fundal al unei pagini web.

A / F

c Dreamweaver este un program de interpretat vise și de creat horoscopul zilnic.

A / F

- Răspundeți la următoarele întrebări:

a Ce este o temă în Google Sites?

b Ce se poate face dacă în timpul editării unei pagini web în Google Sites am șters din greșală un element din pagină care nu trebuia șters?

c Cum putem vedea, la un moment dat, cum arată site-ul la care lucrăm în Google Sites? Cum îl putem vedea la final, înainte de a-l publica pe Internet?

d Care sunt cele două moduri de afișare a iconițelor proiectelor în Google Sites?

e Ce este pagina **Acasă**?

- Scrieți în caiete răspunsul corect pentru următorul enunț.

O temă Google Sites permite:

a aplicarea unei scheme de culori predefinite site-ului vostru;

b adăugarea unor modele 3D site-ului vostru;

c crearea unui site web pornind de la un site web existent.

- Proiect.** Unul dintre avantajele principale ale lui Google Sites este că oferă suport pentru lucrul colaborativ. Scrieți un mic eseu de maximum 200 de cuvinte în care să expuneți părerea voastră legată de cum anume poate profita o instituție școlară de această caracteristică a lui Google Sites și cum anume se poate transforma aceasta într-un veritabil instrument pentru eLearning pentru a ajuta procesul educațional.



Crearea unei pagini web. Publicarea unui site de o pagină



Descoperiți

A Acțiuni pregătitoare

Acum știți deja multe lucruri despre paginile web. Este timpul să facem prima noastră pagină.

Mai întâi va trebui să ne gândim la câteva aspecte, pregătind „terenul“:

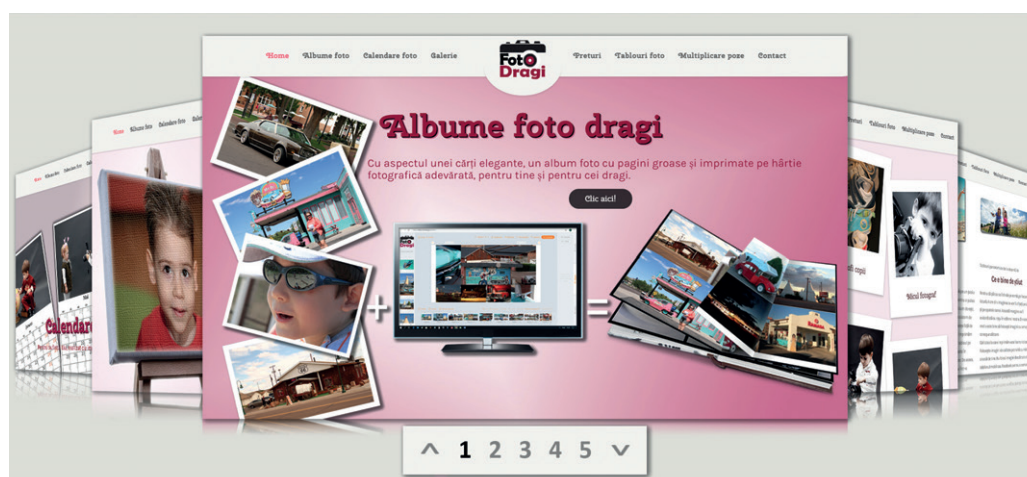
- care va fi tema (subiectul) paginii noastre web;
- ce nume îi vom da;
- care va fi structura paginii (ne gândim ce elemente vrem să punem în pagină și cum vrem să le aranjăm);
- ce conținut va avea fiecare element din pagină (căutăm sau realizăm imaginile, scriem textele, pregătim tabelele etc.).

Ne propunem să facem o pagină web personală pentru un oracol al prietenilor și colegilor noștri. Știți ce este un „oracol“? Conform definiției din *Dicționarul de argou al limbii române* (George Volceanov, 2007), acesta reprezintă o pagină cu un chestionar privind gusturile în materie de muzică, film și altele, dar și cu o predicție legată de viitor, la care răspund colegii de clasă, prietenii etc.

Să creăm, așadar, o pagină dedicată unui astfel de oracol. Îi vom da numele „Oracol + numele propriu“. Pentru exemplificare, să considerăm „Oracol Maria P“.

Drept conținut al paginii ne propunem să avem:

- Carusel de imagini (în antet) – 3, 4 imagini reprezentative pentru noi;



Exemplu de carusel de imagini

- Cuvânt de introducere (poate fi chiar și numai un singur cuvânt);
- Numele;
- Data nașterii/„Când ai aterizat pe Terra?“;
- O imagine cu ceva îndrăgît (obiect, loc, ființă);
- Muzica preferată;
- Filmele preferate;
- Florile preferate;
- Sportul preferat;
- Materiile preferate;
- Secțiunea întrebărilor serioase (inserare paragraf):
 - Ți-ar fi plăcut să trăiești la curtea regală a Franței pe vremea lui Ludovic al XIV-lea? De ce?
 - Ce ai face dacă ai fi multimilionar?

Desigur, putem completa această structură cu orice alte texte, imagini, tabele sau grafice dorim. Putem da frâu liber imaginației.

Pentru variația elementelor pe care le putem introduce într-o pagină web, ne propunem să inserăm și un tabel făcut în Microsoft Excel, de exemplu, un tabel cu punctajele, mediile și clasamentul de la un concurs școlar.

În continuare, pregătim și răspunsurile la toate cele de mai sus. Apoi trecem la pasul următor.

Dacă avem un cont Gmail sau Google, avem deja un cont inclusiv pentru Google Sites. Ne conectăm la contul nostru, alegem un nume de site și o structură de conținut și, în felul acesta, terminăm prima parte.



B Crearea unui site web cu o pagină

Cum putem crea un site în Google Sites? Parcurgeți pașii de mai jos și executați și voi, la rândul vostru, pe calculator:

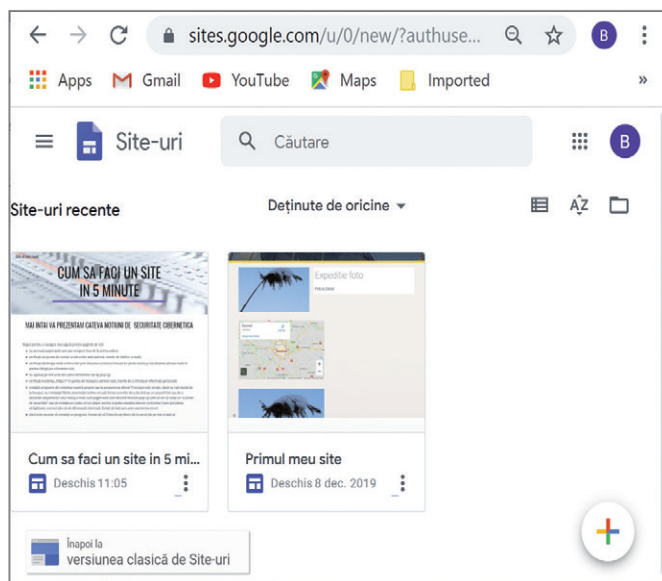
1 **Accesați <https://sites.google.com>.**

2 **Creați un site nou.**

În **Pagina de pornire Site-uri** apăsați butonul + din dreapta-jos (**Creați un site nou**).

3 **Alegeți o temă pentru pagina voastră și configurați antetul.**

Fiecare temă vine cu un fundal presetat, cu o schemă de culori și cu un font selectat. Puteți să modificați oricare dintre aceste elemente, sau chiar tema în sine, în orice moment, chiar și după ce au fost introduse informații pe pagina web. Dacă aveți nevoie să anulați sau să refaceți unele schimbări, puteți folosi butoanele **Undo** (Anulați ultima acțiune) și, respectiv, **Redo** (Refaceți ultima acțiune).



1 – Modificarea imaginii de fundal a antetului:

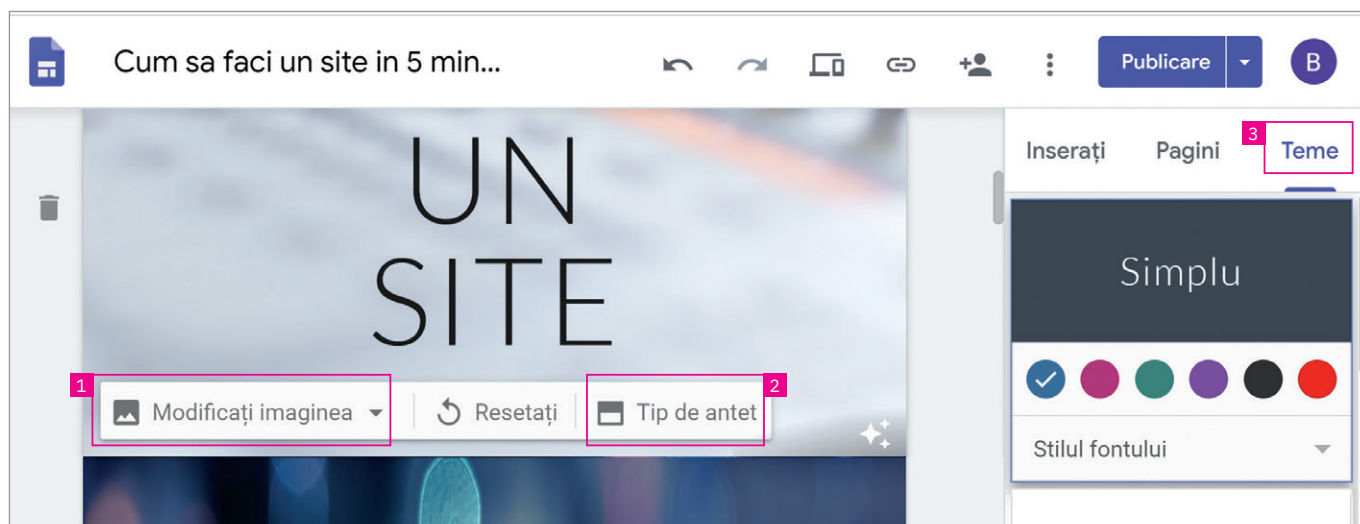
- Mergeți pe zona de antet și selectați **Modificați imaginea**.
- Alegeți una dintre opțiunile:
 - Încărcați o imagine din calculatorul propriu;
 - Alegeți o imagine din galeria de imagini, apoi apăsați pe butonul **Selectați**.
- Pentru a reveni la imaginea de fundal originală, apăsați butonul **Reset** (opțional).

2 – Modificarea tipului de antet:

- Mergeți pe zona de antet și apăsați butonul **Tip de antet**.
- Alegeți o opțiune:
 - Copertă;
 - Banner mare;
 - Banner;
 - Numai titlu.

3 – Schimbarea temei site-ului și a fontului (stilului de scris) folosit:

- Din colțul din dreapta-sus, alegeți **Teme**.
- Selectați o temă care vi se pare mai potrivită și alegeți o culoare (dacă doriți să o modificați pe cea implicită a temei).
- Apăsați pe **Stilul fontului** și alegeți un stil.



4 Dați un nume site-ului și puneți titlu paginii.

Atenție, aveți trei părți diferite ale site-ului care trebuie să primească un nume:

1 – Numele de fișier al site-ului, acel nume cu care îl veți regăsi între fișierele voastre din drive-ul Google. Acest nume este vizibil numai pentru voi, nu și pe Internet.

2 – Numele site-ului. Acest nume apare în antet și în bara de titlu a browserului de Internet după ce faceți site-ul public. Trebuie să aveți două sau mai multe pagini în site-ul vostru web pentru ca acest nume să fie afișat.

3 – Titlul paginii. Fiecare pagină din site-ul vostru trebuie să aibă un nume care apare în partea de sus a paginii. Titlul paginii apare și în meniul de navigare al site-ului.



Numele care apare în bara de titlu a browserului (tagul paginii web) este de ajutor atât pentru utilizator, cât și pentru motoarele de căutare care vor citi și prelucra această informație. Este important ca acesta să fie sugestiv ales și concis formulat pentru a fi clar ce informații se găsesc în pagina respectivă.

Imaginați-vă acum că ați deschis în browserul vostru de Internet mai multe pagini, de exemplu, zece. Cum puteți trece de la o pagină la alta fără să le luați la rând sau pe încercate? Aceasta este o altă situație în care se dovedește utilitatea tagului, pentru că numele site-ului este afișat în eticheta corespunzătoare tabului paginii respective. Când doriți să reveniți la o anumită pagină, o puteți regăsi și accesa cu ușurință datorită numelor site-urilor, care sunt afișate de browserul de Internet, după cum puteți vedea în imaginea de mai jos („Manuale Digitale“, „YouTube“, „Brainly.ro – Rețeaua...“, „Google“).



✓ Rețineți

Câteva recomandări („cele mai bune practici“) pentru a crea taguri bune, sugestive și relevante, atât pentru utilizatori, cât și pentru motoarele de căutare:

- să fie cât mai scurte, ideal ar fi să fie de maximum 55 de caractere;
- nu folosiți numai litere mari;
- nu repetați cuvinte în cadrul unui tag;
- să fie cu adevărat relevante pentru conținutul paginii;
- scrieți nume convingătoare, care v-ar face și pe voi să doriți să faceți clic pe el.

5 Creați paginile și subpaginile dorite (meniul rezultat este disponibil în dreapta-sus); momentan creați o singură pagină, numită **Pagina de pornire** sau, cum este cunoscută în lumea programatorilor, pagina **Acasă**.

✓ Rețineți

O **subpagină web** este o „pagină copil“, o pagină web care este legată de o altă pagină web în structura de pagini a site-ului. Putem avea și subpagini ale unei subpagini.

Odată creată o pagină, se pot alege aspecte pentru secțiunile sale mergând la tabul **Inserați > Aspecte**. Selectarea unui aspect va duce la inserarea unei noi secțiuni în pagina voastră, care are aranjarea imaginilor și textelor corespunzătoare aspectului selectat.

Puteți vedea în imaginea alăturată un exemplu de structură a unei pagini web cu cinci subpagini.

Pagina principală (Acasă)



Exemplu de structură a unei pagini web cu cinci subpagini

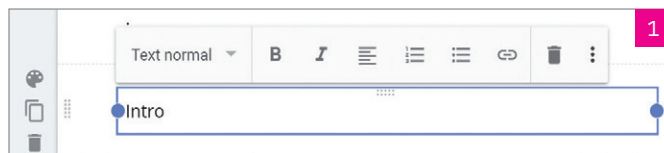
6 Încărcați conținut în pagină: carusel de imagini, texte, poze, hartă (dacă este cazul), casete text etc.

Din colțul din dreapta-sus alegeți **Inserați**. Din bara de unelte care se desfășoară pe verticală puteți insera diferite elemente în pagina web: text (**Casetă text**), text colapsabil (**Text care se poate restrânge**), imagini (**Imagini**), carusel de imagini (**Carousel de imagini**), butoane (**Buton**), separatoare (**Separator**), filme de pe YouTube (**YouTube**), calendare (**Calendar**), hărți (**Hartă**), documente (**Documente**), prezentări (**Prezentări**), foi de calcul (**Foi de calcul**), formulare (**Formulare**), diagrame (**Diagrame**).

Editarea textului

Pentru a edita un text, faceți clic pe căsuța de text și introduceți de la tastatură textul dorit. Puteți, de asemenea, să selectați dintre opțiunile de editare disponibile din caseta de unelte de deasupra (figura 1).

În funcție de stilul ales pentru text (text normal/titlu/antet de rubrică – în engleză, *heading* etc.), veți avea la dispoziție opțiuni diferite de editare/formatare. De exemplu, în imaginea alăturată puteți vedea care sunt opțiunile pentru un text normal.



Descrierea opțiunilor de editare și formatare în funcție de stilul de text este detaliată în tabelul de mai jos.

Pentru titluri și antete de rubrici	
Titlu ▼	Selectați Titlu sau un antet de rubrică presetat. Dacă selectați Text normal , veți avea la dispoziție opțiuni suplimentare de formatare, prezentate în partea a doua a tabelului.
	Schimbați alinierea textului.
	Inserați un hyperlink în textul selectat.
	Ștergeți o căsuță de text.
Pentru text normal	
	Selectați textul dorit și aplicați-i efectul de îngroșat sau aplecat .
	Adăugați liste numerotate sau cu buline.
	Mai multe opțiuni...
	Utilizați stilul de text „Cod de program” – un tip de scris propriu editoarelor folosite de limbajele de programare.
	Selectați textul și strângeți-l.
	Selectați un text și eliminați-i toate opțiunile de formatare aplicate.

Editarea unei secțiuni

Alegeți secțiunea pe care doriți să o editați.

- Schimbați fundalul pentru fiecare secțiune – clic pe **Fundalul secțiunii** și alegeți un stil (Normal, Subliniere 1, Subliniere 2). Încercați, pe rând, toate opțiunile disponibile și alegeți varianta care se potrivește cel mai bine cu proiectul vostru. Puteți să selectați și **Imagine** pentru a adăuga pe fundalul secțiunii o imagine.
- Mutați o secțiune – când introduceți conținut nou, acesta este adăugat la o secțiune nouă sau la una existentă. Pentru a muta o secțiune în sus sau în jos pe pagină, faceți clic pe **Mutare secțiune** și trageți secțiunea unde doriți.
- Duplicarea unei secțiuni – faceți clic pe buton și secțiunea respectivă se va duplica, împreună cu tot conținutul.
- Dacă doriți să ștergeți o secțiune (valabil pentru orice tip de secțiune), e suficient să mergeți la acea secțiune și să faceți clic pe butonul **Ștergeți secțiunea** (figura 2).

O secțiune specială este subsolul paginii. Pentru a adăuga o secțiune de subsol (în engleză, *footer*), mergeți în josul paginii și faceți clic pe **Adăugați subsol**. Secțiunea de subsol va apărea pe toate paginile site-ului vostru. Mergeți la ea și faceți clic pe **Editați subsolul** pentru a o modifica. Pentru a ascunde secțiunea de subsol pe o anumită pagină, deplasați-vă la ea și faceți clic pe **Ascundeți subsolul pe această pagină** .



▶ Editarea imaginilor


• Redimensionarea unei imagini:

Selectați imaginea, apoi trageți de laturile sau de colțurile imaginii. Țineți apăsată tasta **Shift** în timp ce trageți, pentru a păstra raportul laturilor imaginii (pentru a nu o deforma).

• Mutarea unei imagini:

Trageți imaginea într-o altă parte a secțiunii sau într-o nouă secțiune. Puteți plasa imaginea oriunde vedeți o zonă evidențiată.


• Decuparea unei imagini:

– Clic pe imagine, apoi selectați **Decupați** ;

– Trageți cursorul de pe linia orizontală apărută (punctul) pentru a o mări sau a o micșora;


– Puteți mișca imaginea pentru a schimba zona vizibilă;

– Apăsați pe **Editarea s-a terminat** ;

– Pentru a anula decuparea, selectați imaginea și faceți clic pe **Anulați decuparea**  (opțional).

• Înlocuirea unei imagini:

– Selectați imaginea;

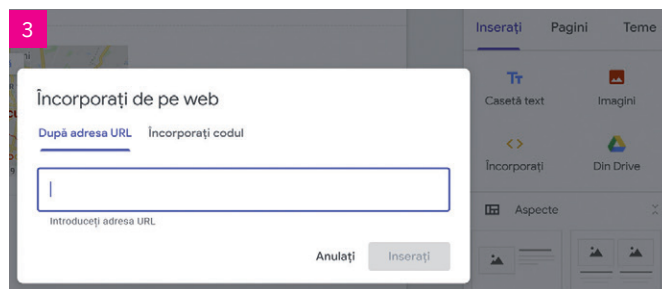
– Clic pe **Mai multe opțiuni de editare**  > **Înlocuiți imaginea** > **Încărcați sau Selectați o imagine** și alegeți un nou fișier imagine.

Adăugarea unui conținut dintr-un alt site web

Puteți încorpora conținut dintr-un alt site web (figura 3).



În funcție de site-ul pe care îl adăugați, puteți personaliza modul în care apare site-ul respectiv pe site-ul nostru. Unele site-uri conțin opțiunea de previzualizare.

- Deschideți proiectul site-ului vostru și din partea dreaptă alegeți **Inserați** > **Încorporați**.
- În caseta care se deschide, introduceți adresa web a site-ului sau a paginii dorite, apoi apăsați **Inserați**.

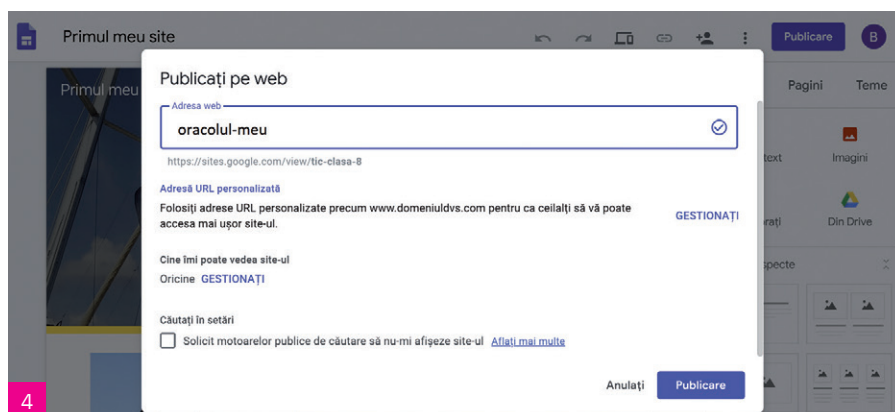



7 Verificați ce a rezultat (butonul Vizualizare).

Dacă vizualizați un site pe un dispozitiv mobil, conținutul se adaptează automat pentru a se potrivi ecranului. Puteți previzualiza cum apare site-ul vostru pe diferite dispozitive (telefon, dispozitiv de tip tabletă sau ecran mare de calculator).

- Clic pe **Vizualizare** .
- Pentru a vedea cum arată site-ul vostru pe diferite dispozitive, alegeți una dintre opțiunile:
 - Telefon;
 - Dispozitiv de tip tabletă;
 - Ecran mare.
- Clic pe **Ieși din modul previzualizare**  pentru a închide acest ecran.

8 Să îl facem public, să îl vadă toată lumea (butonul Publicare). La acest pas trebuie să introducem numele de domeniu dorit. Google ne oferă posibilitatea de a ne găzdui site-ul chiar pe serverele Google. Dacă vom pune numele site-ului „Oracolul-meu“, adresa site-ului nostru va fi <https://sites.google.com/view/oracolul-meu>. Puteți comunica această adresă tuturor prietenilor, rudelor, cunoscuților, pe scurt, faceți site-ul public (figura 4)!



Notă: Vizitați adresa URL a site-ului vostru pentru a verifica dacă site-ul a fost publicat corect. După ce publicați un site, puteți promova site-ul publicat făcând clic pe **Copiați linkul către site-ul publicat**  > **Copiați linkul** și apoi lipiți linkul în mesajele de text sau în e-mailurile pe care le transmiteți.

Și cu această ultimă acțiune, am terminat. Primul nostru site web care conține o pagină este gata și poate fi vizitat online!

**Rețineți**

Când creați un site nou, acesta este adăugat în Google Drive, la fel ca orice alt fișier. Programul salvează automat orice modificare pe care voi o faceți, dar nu este vizibilă pe Internet până nu apăsați butonul **Publicare**.

ȘTIAȚI CĂ...?

Puteți vedea cum sunt realizate diverse pagini web, făcând clic dreapta pe pagina dorită și apăsând **View source page**. Veți vedea codul sursă, adică succesiunea de instrucțiuni HTML pe care calculatorul le interpretează pentru a afișa pagina web așa cum o vedem noi pe ecran.

**Aplicați și investigați****1 Operații cu un tabel**


Inserați în pagina voastră web un tabel creat în Excel și salvat în Google Drive cu filmele preferate și duratele lor, în minute, numit „Filme.xlsx”. Pentru aceasta:

- deschideți pagina voastră și poziționați-vă în locul în care doriți să inserați tabelul;
- mergeți în dreapta, la bara de unelte, și alegeți tabul **Inserați**;
- faceți clic pe **Foi de calcul**;
- din lista care se deschide, faceți clic pe fișierul „Filme.xlsx”;
- apoi clic pe butonul **Inserare**, din partea de jos.

Tabelul creat în Excel este acum inserat în pagina voastră web. Dacă doriți să îi schimbați poziția, doar îl trageți cu mouse-ul în poziția dorită din pagină. Dacă doriți să modificați numele unui film (să editați tabelul), faceți clic pe tabel, apoi clic pe butonul **Deschideți într-o filă nouă** dintre cele trei butoane apărute în stânga-sus. Se deschide, pentru editare, tabelul vostru și puteți face modificarea dorită. După ce ați realizat modificarea și ați închis tabelul, doar apăsați butonul **Reîncărcați această pagină** din browserul în care aveți deschis Google Sites, și modificările făcute în tabel apar imediat.

2 Operații cu o legătură către o altă pagină web

Să presupunem că pe pagina voastră aveți, la rubrica „Muzica preferată”, muzică reggae. Pentru a-i ajuta pe cei care nu cunosc acest gen muzical, vreți să plasați o hiperlegătură către pagina destinată acestui subiect de pe Wikipedia. Astfel:

- în textul vostru, selectați cuvântul „reggae”;
- din meniul care apare deasupra secțiunii de text, faceți clic pe butonul **Inserați un link** ();
- în fereastra care se deschide, introduceți adresa paginii web din Wikipedia (<https://ro.wikipedia.org/wiki/Reggae>);
- apăsați butonul **Aplicați**.

În acest moment, vizitatorii paginii voastre, când citesc acea rubrică, vor putea face clic pe cuvântul marcat (*reggae*) și li se va deschide pagina din Wikipedia destinată muzicii reggae.

Pentru a edita sau pentru a șterge o legătură, doar faceți clic pe textul pe care este aplicată legătura și apăsați butonul **Editați linkul** (pentru modificare) sau **Eliminare** (pentru ștergere).

**Exersați**

- 1 Un „sticker” este prin definiție un „abtbild” autocolant. Prin analogie, un sticker digital este o imagine de dimensiuni mici care poate fi „lipită” peste alte conținuturi digitale (imagini, texte etc.). Faceți cinci stickere și încărcați-le în pagina voastră web. Pentru aceasta puteți utiliza un editor grafic învățat. Există și aplicații online sau aplicații pentru mobile în care le puteți crea; doar căutați „sticker maker”.
- 2 Creați o pagină web (într-un proiect nou) în care să aveți: trei paragrafe de text, patru imagini, un document făcut în Word, un tabel făcut în Excel. Paragrafele de text să fie formate astfel: font Arial de 14, scris normal, unul cu textul aliniat la stânga, unul aliniat la dreapta și unul centrat, iar titlurile să fie îngroșate. Dintre cele trei paragrafe de text, unul trebuie să se repete de două ori. Cum faceți acest lucru fără a munci să îl scrieți și să îl formatați la fel de două ori?
- 3 În pagina web creată la exercițiul 2, formatați textul din secțiunile de text astfel încât acesta să fie centrat și îngroșat.
- 4 Cum se poate face ca fundalul unei secțiuni de text să fie un desen făcut în Paint?
- 5 Scrieți, în caiete, **A** (Adevărat) sau **F** (Fals), în funcție de valoarea de adevăr a propozițiilor:

a	Într-o pagină web făcută de noi putem încorpora conținut dintr-o altă pagină web.	A / F
b	O subpagină nu poate avea și ea o subpagină.	A / F
c	O secțiune de text nu poate avea un fundal colorat într-o singură culoare.	A / F
- 6 **Proiect.** Realizați o pagină web având ca temă sportul vostru preferat și echipa îndrăgită sau sportivul favorit (după caz). Setăți culorile din tema site-ului în concordanță cu culorile echipei îndrăgite. La una din secțiunile de text, fundalul trebuie să fie cu o imagine din sportul respectiv.



Construirea unui site Internet cu mai multe pagini. Lucrul colaborativ



Descoperiți




Mai multe pagini – un site. Site-ul clasei

Scenariul 1: Profesoara/profesorul de la clasă creează un nou proiect de site, cu nume, pagină de pornire și subpaginile aferente: câte o subpagina pentru fiecare elev. Apoi fiecare elev va crea conținutul pentru pagina web alocată, după propria dorință, dar în conformitate cu șablonul stabilit.

Scenariul 2: Profesoara/profesorul de la clasă creează un nou proiect de site doar cu nume și pagină de pornire. Fiecare elev se va conecta la acest proiect, își va crea propria pagină (după un șablon de nume anterior stabilit în comun, de tipul „Oracol_num_e_prenume”), apoi va încărca pagina cu informațiile dorite, respectând însă șablonul stabilit.

Pe oricare dintre aceste scenarii se lucrează, profesorul trebuie să dea drepturi de editare fiecărui elev din clasă (trimite „invitații”), iar voi vă conectați la conturile voastre Google, accesați sites.google.com, deschideți proiectul nou creat, pe care îl regăsiți în lista de proiecte, ca și cum ar fi fost creat de voi. Mergeți la butonul **Pagini** (din colțul din dreapta-sus), deschideți pagina voastră și începeți să o încărcați cu conținut (respectând șablonul stabilit).

Cum se realizează acest lucru? Cum putem face astfel încât să lucrăm împreună, ca o echipă, la același proiect de site web? Iată pașii necesari:

- cineva (profesorul/profesoara, dar poate fi și oricare dintre voi – îl vom numi „lider de grup”) creează proiectul; automat se creează și prima pagină, pagina de pornire; apoi liderul de grup face acțiunile de mai jos;
- clic pe **Trimiteți altor persoane**  ;
- se introduce adresa de e-mail a persoanei invitate și se apasă butonul **Trimiteți**  ;
- se trimite în același mod invitații de acces tuturor persoanelor implicate în proiect;
- la final, se apasă butonul **Terminat**  .

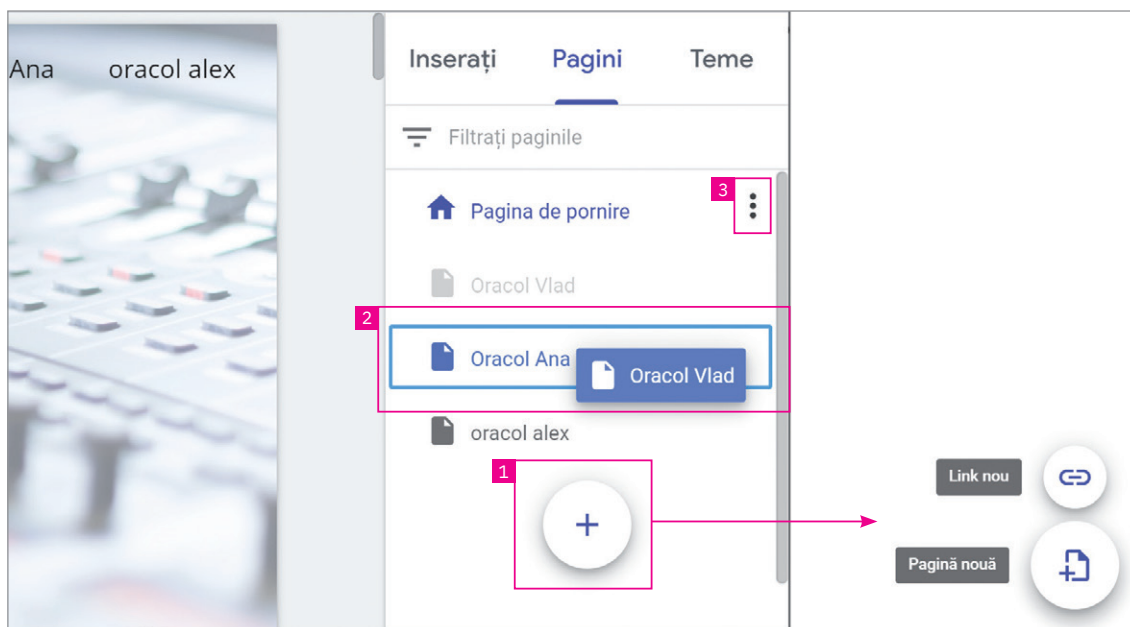


Fiecare persoană care a fost invitată să contribuie la proiect va primi pe e-mail invitația și un link de acces. Clic pe link și lucrul poate să înceapă.

Odată ce avem un site cu mai multe pagini, avem și o serie de noi operații de făcut și opțiuni disponibile. Să le trecem în revistă pe cele mai importante dintre ele.


Adăugați, reordonați și ascundeți paginile

Adăugați pagini pentru mai mult conținut, pentru un site mai bogat. Păstrați informațiile corelate laolaltă prin utilizarea subpaginilor. Subpaginile apar ca un „subtopic” al unei alte pagini, cea principală.



1 – Adăugarea de pagini:

- În colțul din dreapta-sus, faceți clic pe **Pagini > Adaugă** .

Pentru a adăuga o pagină, faceți clic pe **Pagină nouă** . Dați un nume paginii și apoi apăsați **Terminat**.

2 – Reordonarea sau subordonarea paginilor:

- Clic pe **Pagini**;
- Trageți o pagină mai sus sau mai jos în listă pentru a-i schimba ordinea între celelalte;
- Trageți o pagină peste o altă pagină pentru a o subordona. Pagina trasă deasupra va deveni subpagină a celei peste care a fost trasă;
- Pentru a anula subordonarea unei pagini, trageți-o în partea de jos a listei (opțional).

Notă: Putem avea subpagini la subpagini, până la cinci niveluri.

3 – Diferite opțiuni pentru pagini:

De sub **Pagini**, selectați o pagină, în dreptul ei faceți clic pe **Mai multe**  și alegeți una dintre opțiunile:


- **Setați ca pagină principală**;
- **Duplicați pagina**;
- **Adăugați o subpagină**;
- **Ascundeți din bara de navigare**. Nu veți putea ascunde o pagină care este setată ca pagină principală;
- **Ștergeți**. Nu veți putea șterge pagina setată ca pagină principală a site-ului vostru.

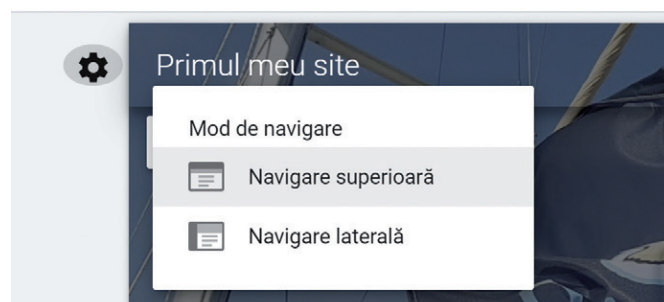
**Configurați navigarea pe site**

Dacă site-ul vostru are mai multe pagini, vizitatorii site-ului vor putea utiliza meniul de navigare pentru a trece de la o pagină la alta. În mod implicit, meniul de navigare se află în partea de sus a site-ului (în colțul din dreapta-sus).

Puteți muta meniul de navigare pe partea stângă (pe verticală) dacă doriți, dar trebuie să aveți mai mult de o pagină pe site-ul vostru pentru a schimba locul unde apare meniul de navigare.


Cum se poate schimba locul în care apare meniul de navigare:

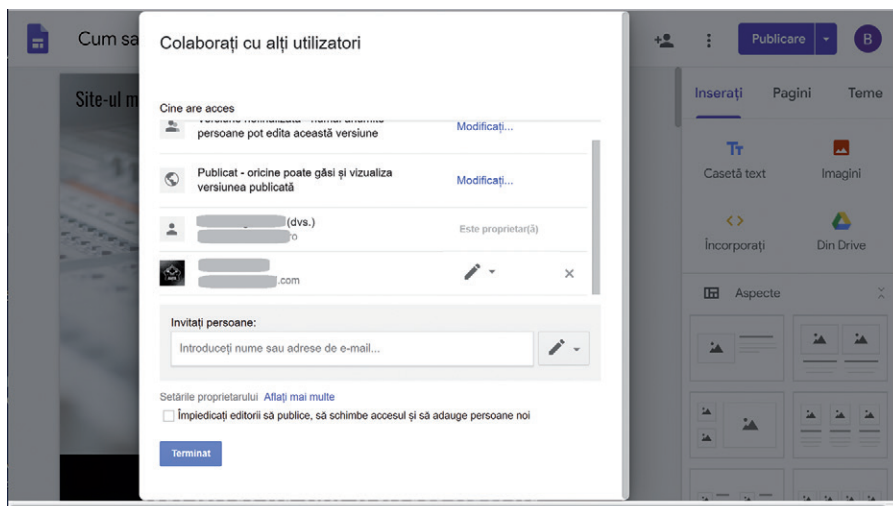
Mergeți în partea de sus, pe zona cu numele site-ului, și faceți clic pe **Setări de navigare** , de unde puteți alege **Navigare superioară** (în partea de sus a paginii) sau **Navigare laterală** (meniul va fi disponibil pe verticală, în partea stângă a site-ului).

**Primul meu site****Partajarea și colaborarea pe Google Sites (cum să lucrăm împreună la un proiect)**

În orice moment al lucrului la site v-ați afla, puteți să îl partajați cu colegii din grupul vostru de lucru pentru a colabora la definitivarea site-ului (și la conținut, și la structură). Am văzut la începutul lecției cum se realizează acest lucru.



Pentru a opri partajarea unui site (oprirea partajării se poate face în orice moment):

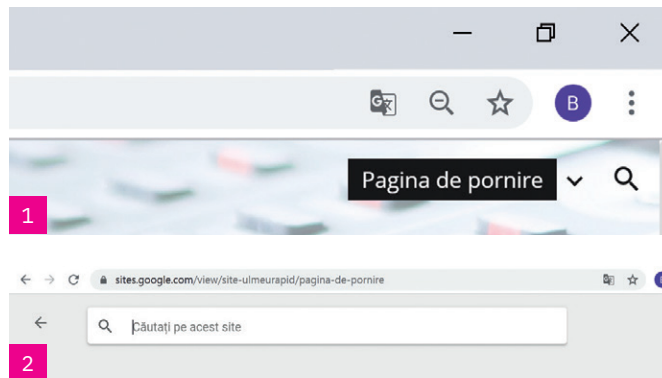
- 1 Deschideți site-ul și, în partea de sus, faceți clic pe **Trimiteti altor persoane**;
- 2 Din lista de sub **Cine are acces**, găsiți persoana pe care vreți să o înlăturați;
- 3 În dreptul numelui respectivei persoane, faceți clic pe **Șterge**  > **Salvați modificările** > **Terminat**.



Căutarea într-un site

Când un site realizat în Google Sites are mai mult de o pagină, aveți la dispoziție o unealtă pentru a căuta un anumit conținut (text). Acest instrument nu este disponibil decât în **varianta publicată** a site-ului.

- 1 Mergeți în colțul din dreapta-sus al site-ului și apăsați **Deschideți bara de căutare**  (figura 1);
- 2 Introduceți textul căutat și apăsați tasta **Enter** sau butonul **Căutare**  din stânga barei (figura 2).




Aplicații și investigații

Realizarea unui site cu cinci pagini, împreună cu un coleg sau un prieten

- 1 Faceți un proiect nou pentru un nou site (accesați <https://sites.google.com> și faceți clic pe butonul + din dreapta-jos);
- 2 Scrieți numele proiectului;
- 3 Mergeți la bara de unelte, la tabul **Pagini**;
- 4 Creați patru pagini (+, **Pagină nouă** și introduceți numele pentru fiecare dintre cele patru pagini);
- 5 Faceți una din pagini să fie subpagină a paginii principale (pagina **Acasă**): trageți respectiva pagină peste pagina căreia i se va subordona;
- 6 Acordați unui coleg drept de editare pe site-ul vostru: faceți clic pe butonul **Trimiteți altor persoane**, introduceți adresa de e-mail a colegului cu care veți lucra la site și apăsați butonul **Trimiteți**. Persoana respectivă va primi pe e-mail invitația și un link de acces. Clic pe acel link, iar el va putea să înceapă lucrul;
- 7 Încărcați (împreună) conținut pe toate cele patru pagini ale site-ului (text, imagini, o foaie de calcul creată în Excel, o listă etc.);
- 8 Apăsați butonul **Vizualizare** pentru a vedea cum va apărea site-ul pentru vizitatori și, când considerați că totul este pus la punct, apăsați butonul **Publicare**. De acum, site-ul vostru este online.



Exersați

- 1 Inițiați un proiect pentru crearea unui site web în Google Sites. Oferiți-i acces unui prieten sau unui coleg pentru a putea lucra și el la acest proiect.
- 2 Dacă aveți create patru pagini în site-ul vostru, cum procedați pentru a face o pagină să devină subpagină a uneia din celelalte trei?
- 3  Completați, în caiete, cuvintele lipsă din textul următor:
Dacă avem un site cu mai multe, vizitatorii site-ului vor putea utiliza meniul de pentru a trece de la o la alta. În mod implicit, meniul de se află în partea de sus a site-ului (în colțul din sus). Putem muta de navigare pe partea stângă (pe), dar trebuie să avem de o pagină pe site pentru a schimba locul unde apare de
- 4 Care dintre opțiunile următoare nu există între opțiunile disponibile pentru o pagină web în Google Sites?
 - a Setări ca pagină principală;
 - b Duplicați pagina;
 - c Adăugați o subpagină;
 - d Mutați pagina;
 - e Ascundeți din bara de navigare;
 - f Ștergeți.
- 5 În Google Sites, odată create paginile unui site web într-o anumită ordine, se mai poate schimba ordinea lor? Dacă da, cum?
- 6 **Proiect.** Împărțiți colectivul clasei în cinci grupuri. Realizați un site web în care să prezentați cinci locuri deosebite din Munții Carpați. Fiecare grup trebuie să pregătească o pagină cu una dintre atracțiile munților noștri. Apoi unificați totul într-un singur site intitulat „Carpații – munții noștri”. Setări culorile folosite în aspectul paginii astfel încât să fie în concordanță cu subiectul ales.

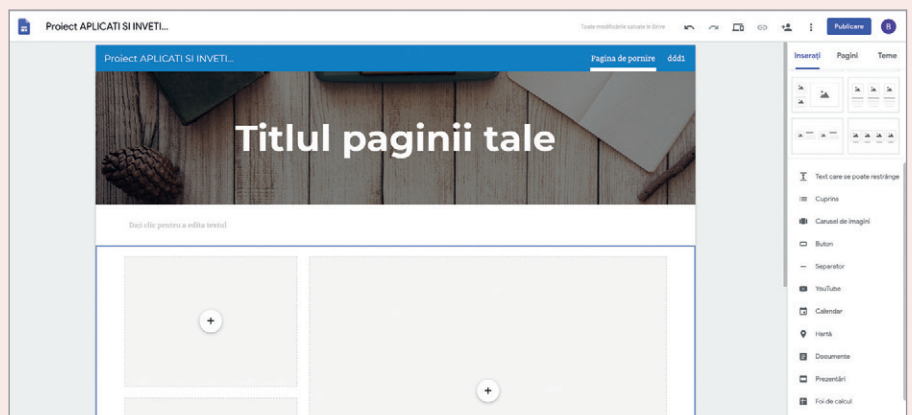
Recapitulare

Rezolvați în caiete sau pe calculator.

- Alegeți varianta corectă. Ce este Google Sites?
 - un limbaj de programare;
 - un program de creat site-uri web;
 - un program de editat texte.
- Care sunt cele patru elemente de structură ale unei pagini web?
- O pagină web poate exista și poate funcționa fără unul dintre aceste trei elemente: antet, corp, subsol?
- Ce reprezintă termenul **slider** în terminologia utilizată în web design?
- Realizați în caiete legătura dintre denumirile următoarelor trei butoane din interfața Google Sites și descrierile corespunzătoare:

a.	Copiați linkul către site-ul publicat	1.	Buton prin care putem invita prieteni sau colegi să lucreze împreună cu noi la același proiect – le dăm acces la proiectul nostru (lucru colaborativ).
b.	Trimiteți altor persoane	2.	Afișează diverse alte meniuri pe care le avem la dispoziție, de la Statistici privind site-ul sau Duplicați site-ul până la Ajutor .
c.	Mai multe	3.	Copiază (în memorie) linkul către site, iar cu ajutorul opțiunii Lipește putem transmite linkul site-ului către oricine dorim.

- Care dintre următoarele acțiuni sunt posibile pentru editarea imaginilor dintr-un site realizat cu Google Sites?
 - Redimensionarea unei imagini;
 - Mutarea unei imagini;
 - Decuparea unei imagini;
 - Înlocuirea unei imagini.
- Alegeți varianta corectă. Locul în care apare meniul de navigare al unui site realizat în Google Sites poate fi setat să apară:
 - în partea de sus a site-ului;
 - pe verticală, în partea dreaptă a site-ului;
 - pe verticală, în partea stângă a site-ului.
- Care dintre următoarele opțiuni nu sunt valabile pentru pagina principală a unui site realizat în Google Sites?
 - Setați ca pagină principală;
 - Duplicați pagina;
 - Adăugați o subpagină;
 - Ascundeți din bara de navigare;
 - Ștergeți.
- Cărui program de realizare pagini web îi aparține interfața din imaginea alăturată?
 - Dreamweaver;
 - Google Sites;
 - Google Web Designer.



- Realizați în Google Sites o pagină web despre rasele de cai românești, folosind tema (șablonul) „Aristotel“.

Autoevaluare

Rezolvați în caiete sau pe calculator.

- 1 Ce este o pagină web?
- 2 Care dintre următoarele elemente pot fi introduse într-un site web?

a tabele;	c pictograme;	e fotografii;
b desene;	d texte;	f modele 3D.
- 3 Realizați, în caiete, corespondența dintre noțiunile date și descrierile lor din tabelul de mai jos:

a. Corpul paginii web	1. acel element care anunță conținutul unei pagini web; îl regăsim, alături de o iconiță sugestivă, în fila browserului corespunzătoare paginii web
b. Antetul paginii web	2. este ultima parte a paginii; poate să conțină câteva informații sumare: un meniu secundar, date de contact și linkuri către platforme sociale
c. Titlul paginii web	3. zona în care găsim informația principală; aceasta este pe primul loc în ceea ce privește spațiul ocupat din pagină
d. Subsolul paginii web	4. zona în care găsim numele paginii și elemente de identitate vizuală

- 4 Alegeți, din lista următoare, aplicațiile dedicate creării de pagini web:

a Dreamweaver;	c MS Word;	e Google Web Designer;
b WordPress;	d Google Sites;	f Wix.
- 5 Explicați pe scurt ce trebuie făcut pentru a putea lucra mai mulți colegi la un site web în Google Sites.
- 6 Numiți două dintre opțiunile pe care le avem la dispoziție pentru modificarea tipului de antet în Google Sites.
- 7 Scrieți, în caiete, **DA** sau **NU** pentru fiecare din afirmațiile de mai jos referitoare la Google Sites:

a Textul editat poate fi: drept (simplu), aplecat, îngroșat, subliniat.	DA / NU
b Textul editat poate să aibă diferite feluri de font, precum și diferite culori.	DA / NU
c Într-un text din pagina web putem insera și o hiperlegătură (hyperlink).	DA / NU
- 8 Completați, în caiete, enunțurile următoare, folosind cuvintele potrivite:
 - a Titlul paginii este de ajutor atât pentru, cât și pentru de căutare care vor citi și prelucra această informație. Este important ca titlul paginii să fie ales și concis formulat pentru a fi clar ce se găsesc în pagina respectivă.
 - b O secțiune specială este paginii. Pentru a adăuga o secțiune de (în engleză *footer*), mergeți în josul paginii și faceți clic pe **Adăugați** Secțiunea de va apărea pe toate paginile site-ului vostru. Mergeți la ea și faceți clic pe **Editați** pentru a o modifica.
- 9 Ce număr maxim de pagini poate conține un site construit cu Google Sites?

a 1;	b 50;	c 100;	d nelimitat.
------	-------	--------	--------------
- 10 Ce sunt și la ce se folosesc cele trei taburi din bara de unelte Google Sites?

Punctaje:

1 – 10 puncte;	6 – 10 puncte;
2 – 10 puncte;	7 – 10 puncte;
3 – 10 puncte;	8 – 10 puncte;
4 – 10 puncte;	9 – 10 puncte;
5 – 10 puncte;	10 – 10 puncte.

Punctaj maxim: 100 puncte.
Timp de lucru: 30 minute.

Răspunsurile se găsesc
la finalul manualului.

Insuficient: 0-50 p
Suficient: 51-60 p
Bine: 61-80 p
Foarte bine: 81-90 p
Excelent: 91-100 p

U3

Algoritmi și programare

- L1 Structura decizională (alternativă) – recapitulare
- L2 Structura repetitivă – recapitulare
- L3 Șiruri de valori (noțiuni introductive)
- L4 Tablou unidimensional. Operații cu șiruri de valori: citire, afișare, parcurgere
- L5 Verificarea unor proprietăți ale elementelor unui șir de valori
- L6 Algoritmi de numărare
- L7 Operații matematice bazate pe valoarea elementelor unui șir
- L8 Căutarea unui element într-un șir
- L9 Sortarea (ordonarea) șirurilor
- L10 Inserarea/ștergerea elementelor unui șir de valori*
- L11 Probleme cu două sau mai multe șiruri
- L12 PROIECT – Utilizarea șirurilor de numere în robotică

Recapitulare

Autoevaluare



Structura decizională (alternativă) – recapitulare



Amintiți-vă

Structura decizională este structura utilizată în cazul în care avem de executat acțiuni diferite în funcție de rezultatul unei decizii (al evaluării unei condiții).



	Pseudocod	Sintaxa C++
Structura decizională (alternativă) cu două ramuri (sunt precizate acțiuni pe două fire de execuție alternative)	<pre> Dacă condiție C Atunci secvență de instrucțiuni A Altfel secvență de instrucțiuni B </pre>	<pre> if (condiție C) { secvență de instrucțiuni A; } else { secvență de instrucțiuni B; } </pre>
Structura decizională (alternativă) cu o singură ramură (sunt precizate acțiuni pe un singur fir de execuție, lipsește alternativa)	<pre> Dacă condiție C Atunci secvență de instrucțiuni A </pre>	<pre> if (condiție C) { secvență de instrucțiuni A; } </pre>
Structura decizională multiplă, cu mai multe ramuri (ramura „Altfel“ poate să lipsească)	<pre> În cazul că expresie selector E Cazul expr_constantă_1: secvență de instrucțiuni 1 [ieșire] Cazul expr_constantă_2: secvență de instrucțiuni 2 [ieșire] Cazul expr_constantă_n-1: secvență de instrucțiuni n-1 [ieșire] [Altfel secvență de instrucțiuni n] </pre>	<pre> switch (expresie selector E) { case expr_constantă_1: secvență de instrucțiuni 1; [break;] case expr_constantă_2: secvență de instrucțiuni 2; [break;] case expr_constantă_n-1: secvență de instrucțiuni n-1; [break;] [default: secvență de instrucțiuni n;] } </pre>



Descoperiți

Scrieți în Code::Blocks programele propuse la punctele **A** și **B** și urmăriți pas cu pas funcționarea lor. Redescoperiți logica de execuție a unui program care include structuri decizionale.

A Analiza informațiilor din CNP (cod numeric personal)

În cadrul unui proces de votare, cartea de identitate a unui alegător este scanată și CNP-ul său este analizat. Realizați un program care citește de la tastatură un număr de 13 cifre reprezentând CNP-ul unei persoane și afișează dacă aceasta este de sex masculin sau feminin și data sa de naștere sub forma AA.LL.ZZ, unde AA sunt ultimele două cifre din an, LL este luna și ZZ este ziua.

Exemplu:

Intrare: n = un număr de forma 2780712XXXXXX, unde X sunt alte cifre

Ieșire: Sex feminin 78 07 12

Indicații:

Pentru a înțelege cum se calculează, priviți forma unui CNP din imaginea alăturată.



Cifrele importante pentru programul de față sunt următoarele:

S	sexul bărbătesc/femeiesc al persoanei; sexul feminin este reprezentat de numerele pare
AA	ultimele două cifre din anul nașterii persoanei
LL	luna nașterii persoanei, în format de două cifre (exemplu: dacă luna nașterii este 5, atunci LL este 05)
ZZ	ziua nașterii persoanei, în format de două cifre (exemplu: dacă ziua nașterii este 7, atunci ZZ este 07)

Pentru rezolvarea problemei, trebuie să extragem prima cifră (pentru a afișa sexul persoanei) și, respectiv, următoarele șase cifre (pentru a afișa data nașterii).

Date de intrare: n = variabilă de tip long long int, reprezintă CNP (număr de 13 cifre)

Date de ieșire: s = variabilă de tip întreg, reprezintă cifra care indică sexul persoanei

z = variabilă de tip întreg, reprezintă ziua nașterii persoanei

l = variabilă de tip întreg, reprezintă luna nașterii persoanei

a = variabilă de tip întreg, reprezintă anul nașterii persoanei

Programul C++ este următorul:

```
#include <iostream>
using namespace std;

int main()
{
    long long int cnp;
    int s, z, l, a;
    cout<<"Introduceți CNP:";
    cin>>cnp;

    s=cnp/1000000000000;
    if (s%2==0) cout<<"Sex feminin ";
    else cout<<"Sex masculin ";

    a=cnp/10000000000%100;
    if (a<10) cout<<"0"<<a<<' ';
    else cout<<a<<' ';

    l=cnp/100000000%100;
    if (l<10) cout<<"0"<<l<<' ';
    else cout<<l<<' ';

    z=cnp/1000000%100;
    if (z<10) cout<<"0"<<z<<' ';
    else cout<<z<<' ';

    return 0;
}
```



Specimen carte de identitate din România



Aplicații și investigații

În programul „Analiza informațiilor din CNP” ștergeți liniile de cod: `if (l<10) cout<<"0"<<l<<' '; else cout<<l<<' ';`. Scrieți în locul lor `cout<<l<<' ';`. La rularea programului pentru un CNP de forma 2780712XXXXX, veți obține ca rezultat: Sex feminin 78 7 12. Care este explicația?



Descoperiți *

B Oracolul din calculatorul tău

Calculatorul poate deveni un adevărat Oracol. Scrieți un program care generează un număr aleator cuprins între 1 și 6. În funcție de valoarea acestuia, afișați câte o „previziune”, un mesaj pentru cel care rulează programul.

Indicații:

Pentru generarea unui număr aleator se utilizează două funcții: `rand()` și `srand()`.

✓ Rețineți



- 1 Funcția `rand()`
`int rand (void);`

Generează un număr pseudoaleator, între 0 și o valoare maximă `RAND_MAX`. `RAND_MAX` este o constantă a cărei valoare diferă în funcție de implementare, dar care este cel puțin 32767.

Pseudoaleator înseamnă că valoarea sa se repetă dacă inițializarea algoritmului de generare s-a făcut pornind de la aceeași valoare inițială („sămânță“).

- 2 Funcția `srand()`
`void srand (unsigned int sămânța);`

Precizează valoarea de inițializare, „sămânță“, utilizată de generatorul de numere al funcției `rand()`.

Dacă „sămânța“ este aceeași (e constantă) la fiecare apelare a funcției `rand()`, atunci numerele aleatoare vor fi generate la fel, urmând aceeași secvență (numere pseudoaleatoare).

Dacă „sămânța“ la inițializare este variabilă, având o valoare diferită la fiecare apelare a funcției `rand()`, atunci se va genera un alt număr.

Implicit, dacă pentru funcția `rand()` nu este precizată nicio „sămânță“ de inițializare, valoarea implicită a acestei „semințe“ este 1.

Exemplu:

Pentru a avea o valoare „sămânță“ diferită la rulări diferite ale programului, se poate utiliza instrucțiunea `srand (time(NULL))`;

Apelul `time(NULL)` returnează numărul de secunde care s-au scurs de la 1 ianuarie 1970. Pentru a utiliza această funcție, trebuie inclusă în program biblioteca `<ctime>`.

Date de intrare: `a` = variabilă de tip întreg, reprezintă numărul generat aleator
Programul în C++ este următorul:



```
#include <ctime>
#include <cstdlib>
#include <iostream>
using namespace std;

int main()
{
    int a;

    //calculatorul alege un numar aleator
    srand(time(NULL));
    a = (rand()%6)+1;

    cout<<"Oracolul spune: ";
    //in functie de numarul aleator se afiseaza un text
    switch (a)
    {
        case 1: cout<<"Vei fi cel mai fericit om din lume!"; break;
        case 2: cout<<"Vei fi foarte bogat!"; break;
        case 3: cout<<"Vei invata sa canti la chitara!"; break;
        case 4: cout<<"Vei intra la liceul dorit!"; break;
        case 5: cout<<"Vei castiga o competitie importanta!"; break;
        case 6: cout<<"O sa intalnesti personal o mare vedeta!";
    }

    return 0;
}
```



ȘTIAȚI CĂ...?

Oracolele jucau un rol foarte important în Grecia antică. Cel mai cunoscut este Templul lui Apollo din Delphi, cunoscut și sub numele *Oracolul din Delphi*. Templul era construit pe o crevasă care se spunea că este o cale către lumea de apoi. Preoteasa Pythia intra în stare de transă și spunea diverse profeții în numele zeilor.



Aplicații și investigații *

- În programul „Oracolul din calculatorul tău” eliminați din cadrul structurii decizionale **switch** toate instrucțiunile **break**; apoi rulați din nou programul. Observați ce se afișează de fiecare dată. Explicați de ce.
- Lucrați în perechi.** Modificați în programul „Oracolul din calculatorul tău” instrucțiunea $a=(\text{rand}()\%6)+1$; în $a=(\text{rand}()\%10)+1$. Care este rezultatul? Completați apoi pe rând mesajele „previziune” până la 10 și rulați programul pentru fiecare membru al echipei în parte. Așa puteți mări diversitatea mesajelor afișate, amuzându-vă în același timp.



Exersați

- În multe aplicații apare necesitatea generării numerelor aleatoare. Dați exemple de două programe care considerați că ar necesita generarea de astfel de numere la întâmplare.
- Realizați un program care calculează suma a două numere întregi mai mici decât 1 000 introduse de la tastatură – dacă numerele sunt divizibile cu 7, respectiv produsul lor – dacă acestea sunt pare. În celelalte cazuri afișați „Alte valori”.
Exemplu: Intrare: 4, 8, Ieșire: 32; Intrare: 7, 14, Ieșire: 21; Intrare: 14, 28, Ieșire: 42 392; Intrare: 3, 8, Ieșire: Alte valori.

- Se introduc de la tastatură valori pentru trei variabile de tip întreg: **a**, **b**, **c**. Scrieți în caiete care dintre seturile de valori prezentate mai jos conduc la afișarea unui număr egal cu valoarea de intrare a variabilei **a** în urma executării secvenței de cod alăturate.

- a** a=26 b=3 c=13
- b** a=26 b=2 c=12
- c** a=15 b=3 c=12
- d** a=26 b=3 c=12

```
int a, b, c;
cin>>a>>b>>c;
if (a>15)
    if (b>2)
        if (c>13) cout<<a%c+b;
        else cout<<a%b*c;
    else cout<<a%b+c;
else cout<<c%b*a;
```



- Care dintre următoarele secvențe de cod afișează valoarea 3, dacă variabila **a** primește la intrare valoarea 3 și variabila **b**, valoarea 4?

a int a, b;
cin>>a>>b;
if (a>5)
 b++;
b--;
cout<<b;

b int a, b;
cin>>a>>b;
if (a>5);
 b++;
b--;
cout<<b;

c int a, b;
cin>>a>>b;
if (a>5);
else b++;
b--;
cout<<b;

- Portofoliu.** Realizați un program care să afișeze dacă trei numere naturale **a**, **b**, **c** pot fi laturile unui triunghi și dacă acesta este dreptunghic/obtusunghic/ascuțitunghic, isoscel/echilateral.

Exemplu:

- Intrare: a=3 b=4 c=5, Ieșire: Numerele pot fi lungimile laturilor unui triunghi. Triunghiul este dreptunghic.;
- Intrare: a=3 b=4 c=4, Ieșire: Numerele pot fi lungimile laturilor unui triunghi. Triunghiul este ascutitunghic, isoscel.;
- Intrare: a=6 b=6 c=6, Ieșire: Numerele pot fi lungimile laturilor unui triunghi. Triunghiul este ascutitunghic, echilateral.;
- Intrare: a=3 b=3 c=7, Ieșire: Numerele nu pot fi lungimile laturilor unui triunghi.

Indicație:

Dacă pătratul celei mai mari laturi a unui triunghi este mai mic decât suma pătratelor celorlalte două laturi, atunci triunghiul este ascuțitunghic, dacă este egal cu suma pătratelor celorlalte două laturi, atunci este triunghi dreptunghic (teorema lui Pitagora) și dacă este mai mare decât suma pătratelor celorlalte două laturi, atunci este obtusunghic.



Structura repetitivă – recapitulare



Amintiți-vă

Există trei tipuri de structuri repetitive:

- 1 Structura repetitivă cu număr necunoscut de pași:
 - a Condiționată anterior;
 - b Condiționată posterior;
- 2 Structura repetitivă cu număr cunoscut de pași.

	Pseudocod	Sintaxa C++
Structura repetitivă condiționată anterior	Cât timp (condiție C) execută secvență de instrucțiuni A 	<code>while (condiție C)</code> <code>{</code> <i>secvență de instrucțiuni A;</i> <code>}</code>
Structura repetitivă condiționată posterior	Repetă secvență de instrucțiuni A Cât timp (condiție C)	<code>do</code> <code>{</code> <i>secvență de instrucțiuni A;</i> <code>} while (condiție C);</code>
Structura repetitivă cu număr cunoscut de pași	Pentru X ← val inițială, val maximă [, pas p] execută secvență de instrucțiuni A 	<code>for (X=val inițială; X<=val maximă;</code> <code>X=X+pas p)</code> <code>{</code> <i>secvență de instrucțiuni A;</i> <code>}</code>



Descoperiți

Scrieți în Code::Blocks programele propuse la punctele **A** și **B** și urmăriți pas cu pas funcționarea lor. Redescoperiți logica de execuție a unui program care include structuri repetitive.

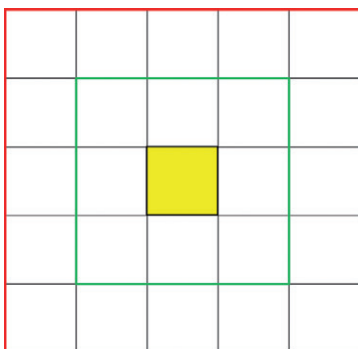
A Problema piramidei

Egiptenii doresc să construiască o nouă piramidă, în trepte. Fiecare nivel al piramidei este construit din blocuri de piatră paralelipipedice cu baza de 1 m x 1 m și înălțimea variabilă în funcție de nivel. Pe primul nivel de jos sunt $n \times n$ blocuri de piatră cu înălțimea de 1 m, unde n este un număr natural diferit de 0. Pe al doilea nivel sunt $(n-2) \times (n-2)$ blocuri de piatră cu înălțimea de 2 m ș.a.m.d. Realizați un program care să citească de la tastatură numărul n și să afișeze numărul de niveluri al piramidei, câte blocuri de piatră se folosesc și înălțimea piramidei în metri.

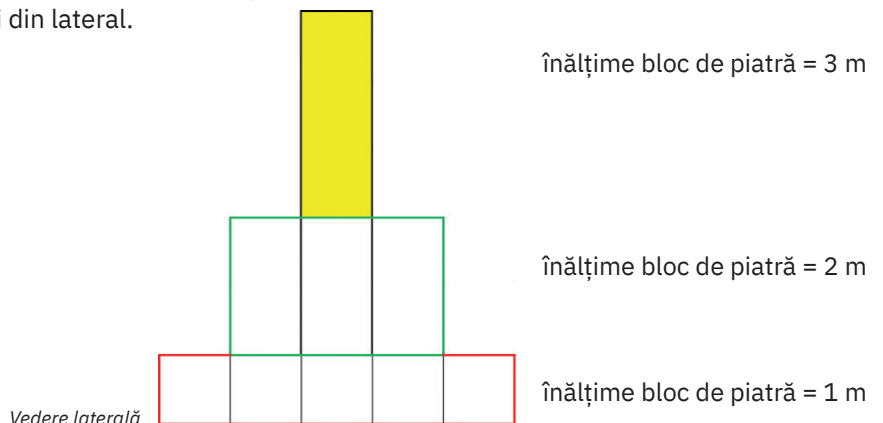
Exemplu:

Intrare: $n=5$; **Ieșire:** Nr. de niveluri este: 3; Nr. blocurilor de piatră folosite este: 35; Înălțimea piramidei este: 6 m

Pentru a înțelege cum anume se calculează rezultatele, priviți imaginile de mai jos ale unei piramide cu baza mare de 5 x 5 blocuri de piatră, văzută de sus și din lateral.



Vedere de sus



Vedere laterală

Indicații:

Date de intrare: n = variabilă de tip întreg, este numărul blocurilor de piatră de pe latura primului nivel de jos

Date de manevră: nrP = variabilă de tip întreg, reprezintă numărul de blocuri de piatră pe un nivel

Date de ieșire: $nrNiv$ = variabilă de tip întreg, reprezintă numărul de niveluri ale piramidei

$nrTP$ = variabilă de tip întreg, reprezintă numărul total de blocuri de piatră folosite

h = variabilă de tip întreg, reprezintă înălțimea în metri a piramidei

Algoritmul se bazează pe faptul că se repetă incrementarea numărului nivelurilor ($nrNiv$), calcularea numărului de blocuri de piatră de pe acel nivel (nrP), a numărului total de blocuri de piatră utilizate până la acel moment ($nrTP$) și a înălțimii piramidei până la acel moment (h), cât timp mai putem să ridicăm un nivel suplimentar ($n - 2 * nrNiv > 0$).

Observați următorul tabel pentru $n=5$:

Nr. nivel	Nr. blocuri pe o latură a nivelului actual	Nr. blocuri previzionat pt. o latură a următorului nivel
1	$5 - 2 * 0 = 5$	$5 - 2 * 1 = 3$
2	$5 - 2 * 1 = 3$	$5 - 2 * 2 = 1$
3	$5 - 2 * 2 = 1$	$5 - 2 * 3 = -1 \Rightarrow$ nu se mai poate construi



Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{   int n, nrNiv=0, h=0, nrP=0, nrTP=0;
    cout<<"n="; cin>>n;
    do
    {
        nrNiv=nrNiv+1;
        nrP=(n-2*(nrNiv-1))*(n-2*(nrNiv-1));
        nrTP=nrTP+nrP;
        h=h+nrNiv;
    } while (n-2*nrNiv>0);
    cout<<"Nr. de niveluri este: "<<nrNiv<<" ";
    cout<<"Nr. blocurilor de piatra folosite este: "<<nrTP<<" ";
    cout<<"Inaltimea piramidei este: "<<h<<" m";
    return 0;
}
```

**ȘTIAȚI CĂ...?**

Cea mai veche piramidă din Egipt este Piramida în trepte din Saqqara sau Piramida lui Djoser. Se spune că a fost realizată de un înalt oficial, pe nume Imhotep, în secolul 27 î.Hr., și se remarcă prin noi standarde de construcție la vremea respectivă și prin stilul arhitectonic deosebit.

**Aplicați și investigați**

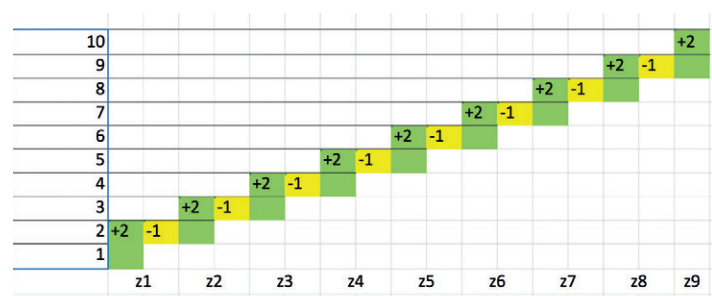
În programul „Problema piramidei” introduceți la intrare un număr par pentru n . Rulați programul pas cu pas și observați câte blocuri de piatră are ultimul nivel în acest caz. Veți vedea că în cazul unui număr n impar veți avea pe ultimul nivel un bloc, iar în cazul unui n par veți avea pe ultimul nivel două blocuri pe o latură. Explicați de ce.

**Descoperiți****B Problema melcului**

Un melc a căzut într-o fântână adâncă de n metri, unde n este un număr natural, $3 \leq n \leq 300$. El încearcă să iasă. Ziua urcă 2 m, iar noaptea cade 1 m. Scrieți un program care afișează în câte zile a ieșit melcul din fântână.

Exemplu: **Intrare:** $n=10$ **Ieșire:** Nr. de zile este: 9

Pentru a înțelege rezultatul, priviți imaginea alăturată în care z_1, \dots, z_9 sunt zilele scurse. În dreptul fiecărei zile figurează înălțimea la care a ajuns melcul.



Indicații:

Date de intrare: n = variabilă de tip întreg, reprezintă adâncimea fântânii

Date de manevră: u = variabilă de tip întreg, reprezintă distanța actuală până la care a urcat melcul

Date de ieșire: z = variabilă de tip întreg, reprezintă numărul de zile scurse până la acel moment

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{   int n, z, u;
    do
    {   cout<<"n= "; cin>>n;
        } while (n<3 || n>100);
    z=1; u=2; u=u-1;
    while (u<n)
    {   z=z+1; u=u+2;
        if (u<n) u=u-1;
        }
    cout<<endl; cout<<"Nr. de zile este: "<<z;
    return 0;
}
```



Aplicații și investigații

Modificați „Problema melcului” astfel încât să introduceți de la tastatură numărul de metri urcați ziua de către melc, du , și numărul de metri coborâți noaptea, dc , unde $dc < du$, iar dc și du sunt numere naturale pozitive. Observați care este corelația dintre diferența $du - dc$ și numărul de zile în care melcul iese din fântână sau între adâncimea fântânii și du , în condițiile în care $du - dc$ este constant.



Descoperiți

c Numere prime

Realizați un program care afișează dacă un număr natural mai mare decât 1 introdus de la tastatură este prim sau nu.

Exemplu:

Intrare: $n=10$, **Ieșire:** Nr. nu este prim; **Intrare:** $n=17$, **Ieșire:** Nr. este prim.

Indicații:

Date de intrare: n = variabilă de tip întreg, reprezintă numărul analizat

Date de manevră: $nrDiv$ = variabilă de tip întreg, reprezintă numărul de divizori diferiți de 1 și de el însuși

Un număr natural n mai mare decât 1 este prim dacă nu are alți divizori diferiți de 1 și de el însuși. Cel mai mic divizor posibil dintre aceștia este 2, deci cel mai mare posibil este $n/2$. Astfel, în algoritm este suficient să încercăm numerele de la 2 până la $n/2$.

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{   int n, i, nrDiv;
    cout<<"n= "; cin>>n;
    nrDiv=0;
    for (i=2; i<=n/2; i++)
        if (n%i==0) nrDiv++;
    if (nrDiv==0) cout<<"Nr. "<<n<<" este prim.";
    else cout<<"Nr. "<<n<<" nu este prim.";
    return 0;
}
```





Aplicații și investigații

În programul „Numere prime“ modificați structura repetitivă `for`, astfel încât să fie `for (i=2; i<=n-1; i++)`. Rulați programul inițial și programul modificat și comparați timpii de execuție.



Exersați

1 Alegeți care dintre secvențele de cod următoare afișează valoarea lui 10!.

a `p=1;`
`for (i=0; i<=10; i++)`
`p=p*i;`
`cout<<p;`

b `p=1;`
`for (i=2; i<=10; i++)`
`p=p*i;`
`cout<<p;`

c `p=1;`
`for (i=2; i=10; i++)`
`p=p*i;`
`cout<<p;`

2 Completați valoarea inițială care trebuie introdusă mai jos pentru a afișa 4 stelute ("****").

```
int k, i;
k=1;
i=.....;
while (i<=10)
{
    k=k*3;
    i=i+2;
    cout<<"*";
}
```

```
int i, l;
l=.....;
i=4;
do
{
    if (i%2==0) {l++; i++;}
    else {i=i+2; cout<<"*";}
} while (i<=l);
```

3 Scrieți, în caiete, A (Adevărat) sau F (Fals) pentru fiecare propoziție de mai jos.

a Secvențele următoare scrise în pseudocod sunt echivalente:

Cât timp (condiție C) execută
 secvență de instrucțiuni A



Dacă (condiție C)
 Atunci

A / F

Repetă
 secvență de instrucțiuni A
 Cât timp (condiție C)



b Secvențele următoare de cod vor afișa același rezultat:

```
int i, j, k=1;
for (i=1; i<=10; i++)
    for (j=1; j<=5; j++)
        k++;
cout<<k;
```

```
int i, j, k=0;
for (i=1; i<=10; i++)
    for (j=1; j<=5; j++)
        k++;
cout<<k+1;
```

A / F

c Secvența alăturată de cod afișează toate numerele naturale pare cuprinse în intervalul [1, 55].

```
int n=55;
while (n>=1)
{
    cout<<n<<' ';
    n=n-2; }

```

A / F

4 Se citesc de la tastatură mai multe numere naturale, până când se introduce valoarea 0. Realizați un program care afișează suma numerelor introduse care conțin exact trei cifre.

Exemplu: **Intrare:** 12, 324, 3, 111, 0; **Ieșire:** 435.

5 Se citesc de la tastatură patru numere naturale. Realizați un program care afișează dacă numărul format din prima cifră a tuturor numerelor introduse este prim sau nu.

Exemplu: **Intrare:** 14, 23, 522, 941; **Ieșire:** Nr. 1259 este prim.

Intrare: 14, 23, 522, 242; **Ieșire:** Nr. 1252 nu este prim.

6 **Portofoliu.** Irina culege mere. Ea culege în prima zi 1 kg de mere, apoi în fiecare zi care urmează culege cu 1 kg în plus față de ziua anterioară. În câte zile va culege Irina necesarul de kilograme de mere pentru piață? Presupunem că la piață merge cu minim 10 kg de mere.

Exemplu: **Intrare:** 10 kg; **Ieșire:** Irina are nevoie de 4 zile in total.



Șiruri de valori (noțiuni introductive)

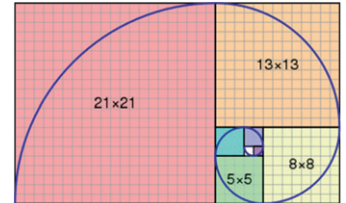


Amintiți-vă

În matematică ați întâlnit până acum noțiunea de șir ca o înșiruire de numere aflate într-o anumită ordine.

Cunoașteți foarte bine, de exemplu, șirul numerelor naturale sau șirul numerelor pare. Un alt șir de numere celebru este *șirul lui Fibonacci*:

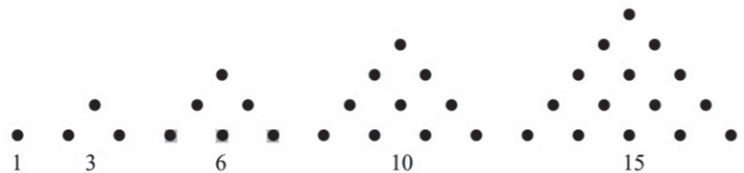
1, 1, 2=1+1, 3=1+2, 5=2+3, 8=3+5, 13, 21, 34, 55, 89, 144, ...



Descoperiți

1 Lucrați în perechi. Desenați cu creioane colorate triunghiurile de mai jos, formate din biluțe. Numărul de bile care formează un triunghi se numește „număr triunghiular”. Dacă numărăm bilele pentru fiecare triunghi începând cu primul, observăm că aceste numere corespund unei reguli bine definite:

Nr. triunghiului	Nr. de bile care formează triunghiul
1	$(1 \cdot 2) / 2 = 1$
2	$(2 \cdot 3) / 2 = 3$
3	$(3 \cdot 4) / 2 = 6$
4	$(4 \cdot 5) / 2 = 10$
...	...
n	$(n \cdot (n+1)) / 2$



Astfel, dacă scriem unul după altul aceste numere, vom obține un șir al „numerelor triunghiulare”. Scrieți în caiete termenul al 50-lea al acestui șir.

2 Închipuiți-vă că aveți de rezolvat o problemă de matematică în care apare un număr foarte mare de numere. Cât timp ar trebui să alocăm rezolvării? Pentru a fi eficienți, este necesar să avem un instrument digital, un program în ajutor. Pentru identificarea câtorva șiruri de numere pe baza unor proprietăți, scrieți în Code::Blocks programele **A**, **B** și **C** propuse mai jos și urmăriți pas cu pas funcționarea lor.

A Șir de numere pare – afișarea primelor n valori

Se dă un număr natural n . Afișați în ordine crescătoare primele n numere naturale pare nenule, unde $1 \leq n \leq 1\,000\,000$.

Exemplu:

Intrare: $n=11$ Ieșire: 2 4 6 8 10 12 14 16 18 20 22

Indicații:

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{   int n, i=1;
    cout<<"n="; cin>>n;
    while (i<=n)
        {   cout<<2*i<<' '; i++; }
    return 0;
}
```

```
n=11
2 4 6 8 10 12 14 16 18 20 22
Process returned 0 (0x0)   execution time : 2.579 s
Press any key to continue.
```

B Șir de numere pare – afișarea valorilor mai mici decât n

Se dă un număr natural n . Afișați în ordine crescătoare numerele naturale pare nenule mai mici sau egale cu n , unde $1 \leq n \leq 1\,000\,000$.

Exemplu:

Intrare: $n=11$ Ieșire: 2 4 6 8 10

Indicații:

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{   int n, i=2;
    cout<<"n="; cin>>n;
    while (i<=n)
        {   cout<<i<<' '; i=i+2; }
    return 0;
}
```

```
n=11
2 4 6 8 10
Process returned 0 (0x0)   execution time : 2.503 s
Press any key to continue.
```



**Aplicați și investigați**

- Inițializați variabila i cu 1 în programul **B**, apoi rulați-l. Ce observați? Ce șir de numere este generat?
- Identificați diferența dintre cerințele problemelor **A** și **B** („primele n ” vs. „mai mici sau egale cu n ”) și urmăriți diferențele de algoritm.
- Modificați programul **A** astfel încât să conțină o structură repetitivă cu număr cunoscut de pași.

**Descoperiți****c Șirul numerelor triunghiulare**

Afișați primele n numere triunghiulare, unde $1 \leq n \leq 100$.

Exemplu:

Intrare: $n=10$ Ieșire: 1 3 6 10 15 21 28 36 45 55

Indicații:

Pentru a afișa șirul numerelor triunghiulare, se aplică formula de calcul a unui element care este prezentată în tabelul de la pagina 66: elementul x al șirului este egal cu $(x*(x+1))/2$.

Programul C++ este următorul:



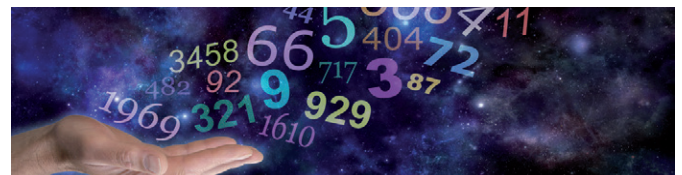
```
#include <iostream>
using namespace std;
int main()
{   int n, i=1;
    cout<<"n="; cin>>n;
    while (i<=n)
    {   cout<<(i*(i+1))/2<<' '; i++; }
    return 0;
}
```

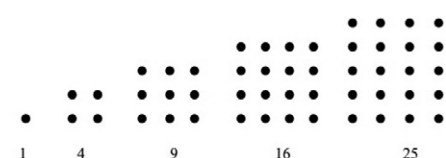
```
n=10
1 3 6 10 15 21 28 36 45 55
Process returned 0 (0x0)   execution time : 4.420 s
Press any key to continue.
```

**Aplicați și investigați**

Aplicați principiul de formare a șirului Fibonacci (elementul x este egal cu suma ultimilor doi predecesori) și generați un nou șir pornind de la șirul numerelor triunghiulare.

Indicație: 1, 4=1+3, 9=3+6, ...

**Exersați**

- Dați exemple de două șiruri de numere ale căror elemente se construiesc după o regulă dată.
- Completați șirurile cu elementele care lipsesc:
 - 1, 4, 7, 10, ..., ..., 19, 22, ..., 28, 31
 - 2, 3, 5, ..., 11, 13, 17, ..., 23, ..., 31, 37, 41, 43, 47, ..., 59
- Se dă șirul de numere: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, ... Care este valoarea elementului al 44-lea al acestui șir?
 - 6;
 - 9;
 - 8;
 - 10.
- Numărul de bile care formează fiecare pătrat în parte din figura alăturată este un „număr pătratic”. Descoperiți algoritmul de formare a șirului numerelor pătratice și faceți un program care să afișeze al n -lea număr pătratic, unde $1 \leq n \leq 100$.
 
- Considerăm șirul numerelor piramidale. Termenul n al acestui șir este egal cu suma primelor n pătrate perfecte, începând de la 1. Scrieți un program care generează primii x termeni ai șirului numerelor piramidale, unde x este număr natural, $1 \leq x \leq 50$.

Indicație: $T_1 = 1^2$, $T_2 = 1^2 + 2^2$, $T_3 = 1^2 + 2^2 + 3^2$, ... $T_x = 1^2 + 2^2 + 3^2 \dots + x^2$

Tablou unidimensional. Operații cu șiruri de valori: citire, afișare, parcurgere



Descoperiți

Lucrul cu șiruri de numere (întregi sau reale) poate presupune un număr foarte mare de elemente și mai multe operații care sunt necesare să fie făcute asupra acestora. Limbajul C++ pune la dispoziția programatorilor posibilitatea de a utiliza șiruri de valori în care putem lucra direct cu fiecare element în parte, acestea putând fi accesate după poziția lor în șir. Astfel, un șir de numere va fi stocat într-o structură de date numită tablou unidimensional.



Rețineți

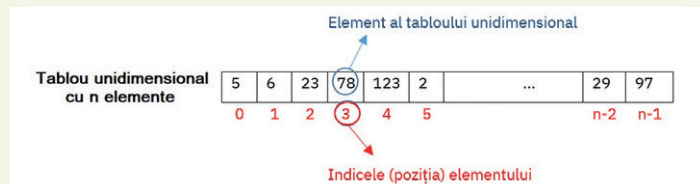
Un **tablou unidimensional** (numit de multe ori **vector**) este o mulțime de valori (elemente) de același tip și care sunt ordonate în funcție de o poziție a lor în cadrul acestuia, numită și indicele valorii respective. Fiecărui tablou unidimensional i se atribuie un **nume**.

Numim **dimensiune** a unui tablou unidimensional numărul maxim de elemente ale sale.

Indicii elementelor unui tablou unidimensional încep de la 0, iar indicele ultimului element al tabloului are valoarea $\langle \text{Dimensiune} \rangle - 1$.

Exemple:

- un tablou unidimensional cu 7 elemente are indicii de la 0 la 6;
- un tablou unidimensional care are n elemente de tip întreg are indicii de la 0 la $n-1$.



A Declaraarea unui tablou unidimensional



Rețineți

Sintaxa în C++ pentru declararea unui tablou unidimensional este:

Tip_date nume[**dimensiune**];

Un tablou unidimensional este o **listă statică** de elemente. La compilare, dimensiunea unui tablou unidimensional trebuie să aibă o valoare clară. În acel moment se va alocă necesarul de memorie pentru toate elementele sale. Odată creat, tabloul va avea pe tot parcursul programului dimensiunea declarată inițial; aceasta nu poate fi schimbată pe parcursul programului, la rulare.

Este foarte important ca la declararea tabloului să estimați dimensiunea sa maximă, dar cât mai mică posibil. Dacă declarați tabloul cu mai multe elemente decât este necesar, atunci se va alocă memorie în plus care nu va fi utilizată niciodată.

Exemplu:

<code>int a[56];</code>	<code>//declararea unui tablou unidimensional de 56 de numere întregi</code>
<code>float b[10];</code>	<code>//declararea unui tablou unidimensional de 10 numere reale</code>
<code>#define DIMENS_MAX 5</code>	<code>//definirea unei dimensiuni maxime constante DIMENS_MAX=5 (directivă preprocesor)</code>
<code>...</code>	
<code>int a[DIMENS_MAX];</code>	<code>//utilizarea constantei DIMENS_MAX pentru declararea tabloului a de numere întregi</code>



Rețineți

În cadrul unei declarații se pot inițializa elementele tabloului cu diferite valori. Sintaxa este:

Tip_date nume[**dimensiune**]={**val_0**, ..., **val_n-1**};

Observație: În momentul în care se face alocarea unui tablou, dacă elementele sale nu sunt inițializate în program, compilatorul nu face niciun fel de inițializare, iar ele vor avea valori oarecare.

Exemple:

```
int v[3]={3,2,4}; //declararea unui tablou cu 3 elemente numere întregi, toate elementele sunt inițializate cu valori
int b[]={5,6,7}; //dimensiunea tabloului nu este precizată, compilatorul o determină numărând elementele din lista
//de valori (în acest caz, dimensiunea este 3)
int c[4]={1,2}; //sunt inițializate primele două valori ale elementelor tabloului, celelalte elemente sunt inițializate cu 0
```



Aplicați și investigați

- 1 Scrieți un program scurt C++ în care introduceți cele 3 declarații de tablouri unidimensionale din exemplele anterioare. Apoi, rulați programul pas cu pas în mod **Depanare (Debug)**, afișați fereastra **Watches** și vedeți exact valorile pe care le au elementele fiecărui tablou în parte (imaginea alăturată).
- 2 Introduceți și declararea unui tablou unidimensional cu 3 elemente întregi (`int v[3];`). Rulați programul în mod **Depanare (Debug)** și observați în fereastra **Watches** că elementele lui `v` au valori oarecare.

Function arguments	
Locals	
v	
[0]	3
[1]	2
[2]	4
b	
[0]	5
[1]	6
[2]	7
c	
[0]	1
[1]	2
[2]	0
[3]	0



Descoperiți

B Referirea unui element al unui tablou unidimensional



Rețineți

Accesarea (referirea) unui element al unui tablou unidimensional se face prin precizarea indicelui său (a poziției sale), astfel:

`nume[indice];`

Parantezele pătrate ([]) sunt operatorul de indexare care are prioritate maximă.

- Exemple:*
- `v[5]` este elementul cu indicele 5 al tabloului cu numele `v`;
 - `v[i]` este elementul cu indicele `i` al tabloului cu numele `v`.

Este important de reținut că atunci când vă referiți la un element al unui tablou unidimensional prin indicele său, trebuie să vă asigurați că indicele este în intervalul corect `[0, X-1]`, unde `X` este dimensiunea sa. Dacă precizați un indice în afara acestui interval, comportamentul programului devine impredictibil.

Dacă începeți lucrul cu elementele unui tablou `v` începând cu indicele 1, atunci elementul cu indice 0 va fi de fapt ignorat. Astfel, dacă doriți să păstrați numărul de elemente utile `X`, declararea tabloului va fi `v[X+1]`.

Exemple:

- dacă aveți un tablou `v` și lucrați cu elemente cu indicii de la 1 la 5, el trebuie declarat `v[6]`;
- dacă aveți un tablou `v` și lucrați cu elemente cu indicii de la 0 la 4, el trebuie declarat `v[5]`.

Un element al unui tablou unidimensional se comportă ca o variabilă care are tipul declarat în program și poate fi utilizat în instrucțiuni/expresii asemenea unei variabile oarecare. Se pot face doar operații asupra elementelor unui tablou, nu se poate modifica tot tabloul deodată.

Exemple: `cin>>v[5]; cout<<v[10]; s=s+v[i]; if (v[i]>=0) {...}`

c Introducerea elementelor unui șir de valori (citirea elementelor șirului)

- 1 Cunoaștem numărul de elemente ale șirului.

Când trebuie să introducem un șir de valori în C++, dacă știm câte elemente are, declarăm un tablou unidimensional cu dimensiunea egală cu numărul de elemente ale șirului și utilizăm operația de citire a valorilor în cadrul unei structuri repetitive.

Exemplificăm introducerea valorilor elementelor unui șir de `n` numere întregi, unde $1 \leq n \leq 100$, într-un tablou unidimensional. Considerăm că primul element are indicele 0.



```
int v[100], n, i; //declararea variabilelor
cout<<"n="; cin>>n; //introducerea numarului de elemente din sir
for (i=0; i<=n-1; i++) //introducerea succesiva a elementelor
{ cout<<"v["<<i<<"]="; cin>>v[i];
}
```

- 2 Nu cunoaștem numărul de elemente ale șirului, dar există o valoare care marchează sfârșitul acestuia la introducere (de obicei, valoarea zero).

Exemplificăm introducerea unui șir de n numere întregi, unde $1 \leq n \leq 100$, la care introducerea valorilor sale se termină când se citește valoarea 0. Valorile introduse se stochează într-un tablou unidimensional v , începând cu indicele 0. Tabloul unidimensional are 100 de elemente.

```
int v[100], a, i=0;           //declararea variabilelor
do                           //se executa cat timp valoarea introdusa nu este 0
{
    cin>>a;                  //citirea valorii
    if (a>0) {v[i]=a; i++;}  //daca valoarea nu este 0, se introduce in tablou
} while (a>0);
```



Aplicați și investigați

În secvența de program de mai sus, inițializați indicele primului element din tabloul v cu 1 ($i=1$). În acest caz, tabloul v va trebui declarat ca având 101 elemente ($v[101]$), deoarece elementul 0 al său va fi ignorat. Investigați și spuneți care este dezavantajul acestui mod de lucru față de cel în care considerăm elementul 0 ca indice al primului element.



Descoperiți

D Parcurgerea elementelor unui șir de valori



Rețineți

Numim **parcurgerea unui tablou unidimensional** accesarea/referirea elementelor sale, unul câte unul, într-o anumită ordine dorită.

Elementele unui tablou pot fi parcurse:

- 1 **de la stânga la dreapta**, adică începând cu primul indice și terminând cu cel mai mare indice din tablou (indicii în ordine crescătoare);
- 2 **de la dreapta la stânga**, adică începând cu cel mai mare indice și terminând cu primul indice (indicii în ordine descrescătoare).

Un exemplu de parcurgere a unui tablou unidimensional este inițializarea elementelor sale cu valoarea 0. Se va utiliza structura repetitivă cu număr cunoscut de pași.

Parcurgere de la stânga la dreapta	Parcurgere de la dreapta la stânga
<pre>int v[100], n, i; ... for (i=0; i<=n-1; i++) v[i]=0;</pre>	<pre>int v[100], n, i; ... for (i=n-1; i>=0; i--) v[i]=0;</pre>

E Afișarea elementelor unui șir de valori

Elementele unui șir stocate într-un tablou unidimensional pot fi afișate **de la stânga la dreapta** sau **de la dreapta la stânga**. Se va parcurge tabloul și se va afișa fiecare element în parte.

Exemplificăm afișarea elementelor pentru un șir de n numere întregi, unde $1 \leq n \leq 100$, utilizând o structură repetitivă cu număr cunoscut de pași. Elementele șirului sunt stocate într-un tablou unidimensional v de dimensiune 100, începând cu poziția 0, și sunt afișate cu un spațiu între ele.



Afișare de la stânga la dreapta	Afișare de la dreapta la stânga
<pre>int v[100], n, i; ... for (i=0; i<=n-1; i++) cout<<v[i]<<' ';</pre>	<pre>int v[100], n, i; ... for (i=n-1; i>=0; i--) cout<<v[i]<<' ';</pre>

Scrieți în Code::Blocks programul propus mai jos și urmăriți pas cu pas funcționarea lui. Descoperiți algoritmi pentru operațiile de bază învățate anterior: citirea, parcurgerea și afișarea elementelor șirului.

F Afișarea elementelor unui șir de valori transformate după o regulă dată

Se citește un șir a de n numere întregi, unde $1 \leq n \leq 250$. Afișați pătratele valorilor elementelor șirului a .

Exemplu: Intrare: $n=5, 3\ 6\ 2\ 8\ 11$ *Ieșire:* $9\ 36\ 4\ 64\ 121$

Indicații:

Deoarece $1 \leq n \leq 250$, definim un tablou a cu 250 de elemente, apoi citim numărul n și cele n elemente ale tabloului. Deoarece nu este specificat în ce ordine să se afișeze pătratele elementelor șirului, considerăm implicit că se vor afișa în ordinea elementelor din șirul a . Astfel, parcurgem șirul a de la stânga la dreapta și afișăm pentru fiecare element al său pătratul calculat, urmat de un spațiu ' '.

În cazul acestei rezolvări, elementele șirului nu se modifică, doar la afișare se aplică formula de transformare (de calcul).

Date de intrare: n = variabilă de tip întreg, reprezintă numărul de elemente din șir

a = tablou unidimensional cu 250 de elemente de tip întreg, reprezintă șirul de numere

Date de manevră: i = variabilă de tip întreg, utilizată drept contor în structura repetitivă

Date de ieșire: a = tablou unidimensional cu 250 de elemente de tip întreg, se afișează elementele sale asupra cărora se aplică formula de calcul (transformarea).

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{   int a[250], n, i;           //definirea variabilelor
    cout<<"n="; cin>>n;       //introducerea numarului de elemente
    for (i=0; i<=n-1; i++)     //citirea secventiala a elementelor tabloului
        cin>>a[i];
    for (i=0; i<=n-1; i++)     //afisarea secventiala a sirului patratelor valorilor
        cout<<a[i]*a[i]<<' ';
}
```

```
n=5
3 6 2 8 11
9 36 4 64 121
Process returned 0 (0x0)   execution time : 6.390 s
Press any key to continue.
```



Exersați

- Scrieți în caiet care sunt valorile elementelor tabloului unidimensional declarat astfel: `int a[6]={0};`.
- Un program care urmează să prelucreze valorile unui șir de 20 de numere reale citește aceste numere într-o variabilă a . Care este declararea corectă a acestei variabile?

a `float a;` **b** `int a[20];` **c** `float a[20];` **d** `float a[19];`

- Se dau două șiruri de numere, $x=\{2, 4, 13, 11, 10\}$ și $y=\{56, 3, 5, 11, 2\}$, care sunt stocate în două tablouri unidimensionale cu același nume, elementele începând de la indicele 0. Completați în caiete valorile următoarelor elemente:

a `y[x[0]]=...` **b** `x[y[1]]=...` **c** `x[y[x[1]]]=...`

- Ce valoare trebuie să primească variabila întregă a pentru ca, în urma executării secvenței de cod alăturate, cel mai mare element al șirului v să aibă valoarea 38?

```
a=...;
for (i=0; i<=5; i++)
    v[i]=a*i+3;
```

- Scrieți secvențele de cod pentru introducerea valorilor unui tablou unidimensional cu n elemente numere întregi, unde $1 \leq n \leq 100$, utilizând structura repetitivă condiționată anterior și, apoi, pe cea condiționată posterior.
- Scrieți un program care să citească un șir de numere naturale pozitive până la introducerea lui 0 și apoi să afișeze elementele acestuia, de la dreapta la stânga, din 2 în 2. Șirul trebuie să aibă maximum 300 de numere.

Exemplu: Intrare: $5\ 2\ 7\ 3\ 1\ 0$ *Ieșire:* $1\ 7\ 5$

- În cadrul unei campanii de marketing, toate prețurile unor produse se reduc cu 20%. Dacă prețurile inițiale dinainte de campanie sunt incluse într-un șir de valori p , iar numărul de produse este n , unde $1 \leq n \leq 150$, afișați lista prețurilor noi din campanie.

Exemplu: Intrare: $n=3\ 7.5\ 10.0\ 20.5$ *Ieșire:* $6\ 8\ 16.4$

- Realizați inversarea ordinii elementelor dintr-un șir de numere întregi cu n elemente, unde $1 \leq n \leq 150$.

Exemplu: Intrare: $n=5\ 1\ 2\ 3\ 4\ 5$ *Ieșire:* $5\ 4\ 3\ 2\ 1$



Verificarea unor proprietăți ale elementelor unui șir de valori



Descoperiți

Scrieți în Code::Blocks programele **A**, **B** și **C** propuse mai jos și urmăriți pas cu pas funcționarea lor. Descoperiți algoritmi prezentați pentru prelucrarea șirurilor de valori în cazul în care este necesară verificarea anumitor proprietăți ale elementelor lor.

A Numere divizibile cu 3

Afișați toate elementele divizibile cu 3 ale unui șir de n numere întregi, unde $1 \leq n \leq 100$. Dacă nu este niciun element divizibil cu 3, atunci nu se afișează nimic.

Exemplu: Intrare: $n=5$ 3 27 9 45 2 *Ieșire:* 3 27 9 45

Indicații:

În cadrul acestui exercițiu, trebuie să afișăm toate elementele șirului care îndeplinesc o condiție dată, adică să fie divizibile cu 3.



Rețineți

Algoritmul pentru afișarea elementelor unui șir de valori care îndeplinesc o condiție este următorul:

- 1 Se citesc elementele șirului.
- 2 Se parcurge șirul și se verifică pentru fiecare element dacă este îndeplinită condiția. Dacă este îndeplinită condiția respectivă, atunci elementul este afișat.

Observație: Condiția este reprezentată de o expresie logică de complexitatea necesară în program.

Șirul de numere îl vom stoca într-o variabilă v de tip tablou unidimensional cu dimensiunea 100, acesta fiind numărul maxim posibil de elemente ($1 \leq n \leq 100$). Un număr X este divizibil cu 3 când restul împărțirii lui la 3 este 0, deci condiția de divizibilitate cu 3 este $X \% 3 == 0$. Se vor afișa elementele șirului care îndeplinesc această condiție.

Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{
    int v[100], n, i;
    cout<<"n="; cin>>n;           //se introduce numarul de elemente din sir

    for (i=0; i<=n-1; i++)       //se introduc elementele sirului
        cin>>v[i];

    for (i=0; i<=n-1; i++)       //se parcurge sirul de la stanga la dreapta
        if (v[i]%3==0)           //se verifica indeplinirea conditiei de catre elementul i
            cout<<v[i]<<' ';     //se afiseaza elementul care indeplineste conditia

    return 0;
}
```

```
n=5
3 27 9 45 2
3 27 9 45
Process returned 0 (0x0)   execution time : 16.335 s
Press any key to continue.
```



Aplicați și investigați

- 1 Modificați programul anterior astfel încât să afișeze toate numerele pare divizibile cu 3 ale șirului. În acest caz, condiția de afișare devine $(v[i]\%3==0 \ \&\& \ v[i]\%2==0)$.
- 2 Modificați programul de mai sus astfel încât să se afișeze pozițiile elementelor divizibile cu 3, nu elementele în sine. Observați că, în acest caz, în loc să afișăm $v[i]$, trebuie afișat doar i .



Descoperiți

B Maximul unui șir de numere

Aflați maximul dintr-un șir de n numere naturale pozitive, introduse de la tastatură, mai mici decât 1 000 000, unde $1 \leq n \leq 20$.

Exemplu: **Intrare:** $n=5$ 21 54 3 23 5 **Ieșire:** Maximul=54

Indicații: În cadrul acestui exercițiu trebuie să afișăm o singură valoare, valoarea maximă a elementelor șirului. Este un caz particular al afișării elementelor care îndeplinesc condiția dată, și anume valoarea elementului să fie cea mai mare a șirului.



Rețineți

Algoritmul pentru afișarea maximului unui șir de numere este următorul:

Se consideră o variabilă pe care o numim **max**, care va conține la fiecare pas valoarea maximă a elementelor șirului analizate până la acel moment, variabilă de ieșire.

- 1 Se citesc elementele șirului.
- 2 Se parcurge șirul și se verifică pentru fiecare element în parte dacă valoarea acestuia este mai mare decât valoarea variabilei **max** (aceasta reprezintă maximul depistat până la acel moment). Dacă da, atunci noul maxim este chiar elementul analizat și, astfel, **max** primește valoarea sa.
- 3 Se afișează valoarea variabilei **max** care conține maximul șirului.

Observație: Variabila **max** se inițializează cu o valoare astfel încât aceasta să nu influențeze rezultatul. Această valoare poate fi: 1) o valoare mai mică decât orice element; sau 2) valoarea elementului de la care se începe parcurgerea șirului pentru verificare.

Algoritmul pentru **determinarea minimului unui șir de numere** este similar. Diferă variabila de ieșire, care de această dată va reține minimul (numită **min**) și condiția verificată pentru fiecare element (dacă valoarea elementului este mai mică decât valoarea variabilei **min**, atunci noul minim este chiar elementul respectiv).

Vom stoca șirul de numere într-o variabilă **v** de tip tablou unidimensional cu 20 de elemente, 20 fiind valoarea maximă pe care o poate lua n ($1 \leq n \leq 20$). Se aplică algoritmul de mai sus pentru calcularea maximului.



Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{
    int v[20], n, i, max=0;
    cout<<"n="; cin>>n;           //se citește numărul de elemente ale șirului
    for (i=0; i<=n-1; i++)       //se citesc elementele șirului
        cin>>v[i];
    for (i=0; i<=n-1; i++)       //se parcurge șirul
        if (max<v[i])           //se verifică pentru fiecare element dacă este mai mare
            max=v[i];           //dacă da, maximul devine chiar elementul
    cout<<"Maximul="<<max;      //se afișează maximul
    return 0;
}
```

```
n=5
Sirul: 21 54 3 23 5
Maximul=54
Process returned 0 (0x0)   execution time : 5.825 s
Press any key to continue.
```



Aplicați și investigați

- 1 Modificați programul de mai sus în așa fel încât să se afișeze indicele elementului maxim. Pentru aceasta, declarați o variabilă nouă întregă **iM**, inițializată cu -1 (**int iM=-1**), în care rețineți de fiecare dată indicele **i** al elementului șirului a cărui valoare este găsită a fi mai mare decât valoarea lui **max** (când **max<v[i]**, executați și **iM=i**). Rulați programul în mod **Depanare** (*Debug*) și urmăriți ce se întâmplă cu valoarea variabilei **iM**.
- 2 Modificați programul anterior astfel încât variabila **max** să fie inițializată cu primul element al șirului. Observați că, în acest caz, numărul de verificări dacă (**max<v[i]**) este mai mic cu 1. Secvența de verificare este cea alăturată.

```
max=v[0];
for (i=1; i<=n-1; i++)
    if (max<v[i]) max=v[i];
```



Descoperiți

c Înlocuirea unor termeni ai unui șir de valori

Înlocuiți elementele unui șir care au valoarea 7 cu valoarea 0. Șirul are n numere întregi, unde $1 \leq n \leq 100$.

Exemplu: Intrare: $n=4$ 2 4 7 1 Ieșire: 2 4 0 1

Indicații:

În cadrul acestui exercițiu, trebuie să verificăm existența, într-un șir de valori, a unui termen care are o proprietate dată, și anume să fie egal cu 7. Dacă îl găsim, îl înlocuim cu valoarea 0.



Rețineți

Algoritmul pentru înlocuirea termenilor unui șir de valori care îndeplinesc o condiție este următorul:

- 1 Se citesc elementele șirului.
- 2 Se parcurge șirul și se verifică pentru fiecare element dacă este îndeplinită condiția.
- 3 Dacă este îndeplinită condiția respectivă, atunci elementul este înlocuit cu noua valoare.
- 4 Este afișat șirul modificat.

Observație: Condiția este reprezentată de o expresie logică, de complexitatea necesară în program.



Programul C++ este:

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    int v[100], n, i;
    cout<<"n="; cin>>n;           //se citeste numarul de elemente ale sirului
```

```
    for (i=0; i<=n-1; i++)       //se citesc elementele sirului
        cin>>v[i];
```

```
    for (i=0; i<=n-1; i++)       //se parcurge sirul de la stanga la dreapta
        if (v[i]==7)             //daca este indeplinita conditia (egalitatea cu 7)
            v[i]=0;              //se inlocuieste valoarea elementului cu 0
```

```
    for (i=0; i<=n-1; i++)       //se afiseaza sirul modificat
        cout<<v[i]<<' ';
```

```
    return 0;
```

```
}
```

```
n=4
2 4 7 1
2 4 0 1
Process returned 0 (0x0)   execution time : 12.475 s
Press any key to continue.
```



Aplicați și investigați

1 Modificați programul de mai sus astfel încât să se înlocuiască cu 0 toate numerele pozitive nenule divizibile cu ultimul element din șir. În acest caz, condiția devine $(v[i]\%v[n-1]==0 \ \&\& \ v[i]>0)$.

2 Rulați programul de mai sus și notați timpul de rulare. Înlocuiți primele două structuri repetitive `for` cu structura compactată alăturată care duce la același rezultat. Rulați din nou programul pe același set de date ca înainte și notați timpul. Observați că timpul de rulare a scăzut. Investigați și explicați de ce.

```
for (i=0; i<=n-1; i++)
{
    cin>>v[i];
    if (v[i]==7)
        v[i]=0;
}
```



```
if (v[i]==X)
    cout<<"Imi place C++!";
```



Exersați

1 Se dă secvența de cod alăturată. Scrieți în caiete care este varianta corectă dintre cele de mai jos:

- a se afișează elementele impare de pe pozițiile impare ale tabloului v ;
- b se afișează elementele pare de pe pozițiile impare ale tabloului v ;
- c se afișează elementele impare de pe pozițiile pare ale tabloului v .

```
for (k=30; k>=0; k--)  
    if (v[k]%2==0 && k%2==1)  
        cout<<v[k]<<' ';
```

```
for (i=0; i<=8; i++)  
    if (i%3==0)  
        cout<<v[i]+2*i;
```

2 Tabloul unidimensional v conține elementele 5, 2, 7, 1, 3, 4, 3, 2, 9, începând cu indicele 0. Scrieți în caiete ce se va afișa pe ecran în urma rulării secvenței de cod alăturate.

3 Calculați maximul unui șir de numere după exemplul prezentat în lecție, dar inițializând variabila max cu ultimul element al șirului. Șirul are n numere naturale pozitive mai mici decât 1 000 000, unde $1 \leq n \leq 100$.

4 Dați un exemplu de aplicabilitate practică a algoritmului de calculare a maximului unui șir de numere.

5 Marcel este un om foarte calculat. În decursul unui an, el știe care a fost totalul cheltuielilor sale în fiecare lună în parte. În nicio lună nu a cheltuit mai mult de 1 500 RON. Ajutați-l să facă un program care să îi afișeze luna în care a cheltuit cel mai puțin.

Exemplu: **Intrare:** 1234 1432 1222 1001 978 987 1223 1455 1500
1378 1226 1190 **Ieșire:** Luna mai

6 Se consideră un șir de maxim 100 de numere întregi nenule introdus de la tastatură, care începe de la poziția 1, iar citirea valorii 0 marchează sfârșitul său. Să se afișeze:

- a toate elementele care sunt situate pe poziții pare, dacă există;
- b toate elementele situate pe poziții impare, afișate de la dreapta la stânga, dacă există;
- c pozițiile tuturor elementelor divizibile cu 5 care sunt pare, dacă există.

Exemplu: **Intrare:** 60 4 3 2 78 3 0 **Ieșire:** a: 4 2 3 b: 78 3 60 c: 1

7 Afișați toate elementele unui șir de n numere naturale, unde $1 \leq n \leq 100$, care sunt divizibile cu suma ultimelor două elemente.

Exemplu: **Intrare:** n=5 2 12 3 2 1 **Ieșire:** 12 3

8 Un lanț de supermarketuri are 6 magazine, numerotate de la 1 la 6. Fiecare magazin are un număr de maximum 10 000 de produse. Realizați un program care să afișeze magazinele în care produsele pot fi împachetate în cutii cu o capacitate de 50 de produse, fără a mai rămâne vreun produs în plus.

Exemplu: **Intrare:** 2300 7200 5324 3748 9867 8550 **Ieșire:** Magazinele: 1, 2, 6

9 Scrieți un program care să înlocuiască toate elementele prime ale unui șir de n numere întregi cu dublul lor ($1 \leq n \leq 100$).

Exemplu: **Intrare:** n=5 2 13 45 11 72 **Ieșire:** 4 26 45 22 72

10 Scrieți un program care să calculeze maximul dintr-un șir de n numere naturale pozitive mai mici decât 1 000 000, unde $1 \leq n \leq 100$, și apoi să înlocuiască toate elementele pare cu acesta.

Exemplu: **Intrare:** n=5 3 4 5 6 7 **Ieșire:** 3 7 5 7 7

11 Pentru a se cripta un număr de 10 cifre, toate cifrele sale se înscriu într-un tablou unidimensional, primul element fiind pe poziția 0. Asupra fiecărei cifre se aplică un algoritm de transformare astfel: fiecare cifră se înlocuiește cu ultima cifră a sumei dintre ea și indicele ei din tablou. Afișați numărul criptat rezultat.

Exemplu: **Intrare:** 6789243596 **Ieșire:** 6802699275

12 Să se afișeze dacă minimul și maximul unui șir de n numere naturale, unde $1 \leq n \leq 100$, sunt prime între ele.

Exemplu: **Intrare:** n=6 23 2 45 34 46 **Ieșire:** Nu

Indicație: Minimul este 2, maximul este 46, c.m.m.d.c. (2,46)=2



Algoritmi de numărare



Descoperiți

Scrieți în Code::Blocks programele **A**, **B** și **C** propuse mai jos și urmăriți pas cu pas funcționarea lor. Veți descoperi noi algoritmi pentru obținerea de rezultate dintr-un șir de valori, de data aceasta pentru numărarea elementelor în diferite cazuri.

A Verificarea existenței unor elemente cu anumite proprietăți

Să se verifice dacă într-un șir de n numere întregi pozitive, unde $1 \leq n \leq 100$, există numere pare.

Exemplu: Ințrare: $n=4$ **2 3 1 6** Ieșire: Da; Ințrare: $n=4$ **7 3 1 5** Ieșire: Nu

Indicații:

În acest exercițiu verificăm existența într-un șir de valori a unor termeni cu o anumită proprietate și afișăm **Da/Nu**.



Rețineți

Algoritmul pentru afișarea existenței unor elemente ale unui șir cu o anumită proprietate este următorul:

Se consideră o variabilă întreagă **eNr** care poate lua două valori: 0 – dacă nu există niciun element cu proprietatea dată, 1 – dacă există minim un element cu proprietatea dată. Variabila se va inițializa cu 0, valoarea ei schimbându-se în 1 cu ocazia găsirii primului element cu proprietatea dată.

- 1 Se citesc elementele șirului și se inițializează variabila **eNr** cu 0.
- 2 Se parcurge șirul și se verifică fiecare element dacă are proprietatea cerută. Dacă este îndeplinită condiția respectivă, atunci **eNr=1**.
- 3 Dacă **eNr=1**, se afișează **Da**, altfel se afișează **Nu**.

Vom stoca șirul de numere într-un tablou unidimensional **v** cu 100 de elemente ($1 \leq n \leq 100$). Se aplică algoritmul de mai sus pas cu pas. Un element este par dacă restul împărțirii la 2 este 0 ($v[i]\%2==0$).

Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{   int v[100], i, n, eNr=0;
    cout<<"n="; cin>>n;           //se citește numărul de elemente ale șirului
    for (i=0; i<n-1; i++)         //se citesc elementele șirului
        cin>>v[i];
    for (i=0; i<n-1; i++)         //se parcurg elementele șirului
        if (v[i]%2==0) eNr=1;    //valoarea lui eNr devine 1 dacă elementul este par
    if (eNr==1) cout<<"Da";
    else cout<<"Nu";
    return 0;
}
```

```
n=4
2 3 1 6
Da
Process returned 0 (0x0)   execution time : 2.795 s
Press any key to continue.
```

```
n=4
7 3 1 5
Nu
Process returned 0 (0x0)   execution time : 2.536 s
Press any key to continue.
```



Aplicați și investigați

- 1 Înlocuiți ultima secvență repetitivă **for** cu o secvență repetitivă condiționată posterior, după cum puteți vedea alături. Rulați și descoperiți diferențele dintre cele două soluții.

```
i=0;
do
{   if (v[i]%2==0) eNr=1;
    else i++;
} while (i<n-1 && eNr==0);
```

- 2 Investigați posibilitatea de a rezolva problema: *Să se verifice dacă într-un șir de n numere întregi pozitive, unde $1 \leq n \leq 100$, toate numerele sunt pare.* Aplicați algoritmul de mai sus, dar inițializați variabila **eNr** cu 1 (presupuneți că toate numerele sunt pare). Veți schimba apoi valoarea lui **eNr** în 0 doar dacă veți întâlni un număr impar.

**Descoperiți****B Numărarea elementelor care au o proprietate anume**

Afișați câte numere pare sunt într-un șir de n numere întregi pozitive, unde $1 \leq n \leq 100$.

Exemplu: Intrare: $n=4$ 2 3 1 6 *Ieșire:* Nr pare: 2

Indicații: Pentru a rezolva problema trebuie să numărăm elementele șirului care îndeplinesc condiția de paritate.

**Rețineți**

Algoritmul pentru numărarea elementelor unui șir care au o proprietate anume este următorul:

Se consideră o variabilă întregă nr care va reține numărul elementelor pare (este un contor).

- 1 Se citesc elementele șirului și se inițializează variabila nr cu 0.
- 2 Se parcurge șirul și se verifică pentru fiecare element dacă are proprietatea cerută. Dacă este îndeplinită condiția respectivă, atunci se incrementează valoarea contorului nr .
- 3 Se afișează valoarea variabilei nr .

Vom stoca șirul de numere într-un tablou unidimensional v cu 100 de elemente, pentru că $1 \leq n \leq 100$.

Programul C++ este:



```
#include <iostream>
using namespace std;
int main()
{
    int v[100], n, i, nr=0;
    cout<<"n="; cin>>n;           //se citește numărul de elemente ale șirului
    for (i=0; i<=n-1; i++)       //se citesc elementele șirului
        cin>>v[i];
    for (i=0; i<=n-1; i++)       //se parcurge șirul
        if (v[i]%2==0) nr++;     //se numără elementele pare
    cout<<"Nr pare: "<<nr;       //se afișează numărul de elemente pare ale șirului
    return 0;
}
```

```
n=4
2 3 1 6
Nr pare: 2
Process returned 0 (0x0)   execution time : 2.235 s
Press any key to continue.
```

**Aplicați și investigați**

- 1 Investigați și găsiți și alte metode de a rezolva problema anterioară **A**, folosind algoritmul de numărare. Determinarea existenței unui element poate fi privită ca un caz particular al numărării. Putem rezolva acest algoritm și numărând numerele pare întâlnite în șir. Dacă la numărătoare obținem 0, atunci nu există niciun număr par.
- 2 Compactați cele două structuri repetitive așa cum este prezentat în secvența de cod alăturată. Rulați ambele versiuni de program pe mai multe seturi de date și comparați timpul de execuție. Ce observați?

```
for (i=0; i<=n-1; i++)
{
    cin>>v[i];
    if (v[i]%2==0) nr++;
}
```

**Descoperiți****c Statistică**

Se dă un șir de maximum 100 de numere naturale cu valori cuprinse între 1 și 5. Să se afișeze procentul elementelor egale cu 1, cu 2, cu 3, cu 4 și cu 5.

Exemplu: Intrare: $n=10$ 1 2 5 1 2 3 2 4 5 5

Ieșire: % valoare 1: 20%, % valoare 2: 30%, % valoare 3: 10%, % valoare 4: 10%, % valoare 5: 30%

Indicații:

Pentru rezolvarea exercițiului, aplicăm algoritmul B – Numărarea elementelor care au o proprietate anume pentru fiecare tip de element căutat (egal cu 1, 2, 3, 4 și, respectiv, 5). Apoi calculăm procentul elementelor de fiecare tip după formula: $\langle \text{Număr elemente care îndeplinesc condiția dată} \rangle / \langle \text{Număr total de elemente în șir} \rangle * 100$.

Procentul este un număr real.

Vom stoca șirul de numere într-un tablou unidimensional v cu 100 de elemente, pentru că $1 \leq n \leq 100$.



Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{
    int v[100], n, i;
    int nr1=0, nr2=0, nr3=0, nr4=0, nr5=0;
    cout<<"n="; cin>>n;           //se citește numărul de elemente ale șirului
    for (i=0; i<n-1; i++)       //se citesc elementele șirului
    {
        cin>>v[i];
        if (v[i]==1) nr1++;
        if (v[i]==2) nr2++;
        if (v[i]==3) nr3++;
        if (v[i]==4) nr4++;
        if (v[i]==5) nr5++;
    }
    //se afișează procentul elementelor 1, 2, 3, 4, 5
    cout<<"% valoare 1: "<<(float)nr1/n*100<<"%"<<endl;
    cout<<"% valoare 2: "<<(float)nr2/n*100<<"%"<<endl;
    cout<<"% valoare 3: "<<(float)nr3/n*100<<"%"<<endl;
    cout<<"% valoare 4: "<<(float)nr4/n*100<<"%"<<endl;
    cout<<"% valoare 5: "<<(float)nr5/n*100<<"%"<<endl;

    return 0;
}
```

```
n=10
1 2 5 1 2 3 2 4 5 5
% valoare 1: 20%
% valoare 2: 30%
% valoare 3: 10%
% valoare 4: 10%
% valoare 5: 30%

Process returned 0 (0x0)   execution time : 3.704 s
Press any key to continue.
```



Aplicații și investigații

Activitate de grup. Utilizarea practică a acestui algoritm este pentru calculul statistic al răspunsurilor la anumite întrebări din cadrul unor cercetări bazate pe aplicarea de chestionare. Realizați un sondaj de opinie la voi în clasă legat de liceele preferate ale fiecărui elev. Completați cu câte licee doriți. Apoi, introduceți datele în programul anterior și vedeți rezultatul! Care este liceul preferat de cei mai mulți colegi? Puteți extinde cercetarea la toți elevii din clasa a VIII-a din școala voastră, de exemplu.

Întrebarea este: Cât de mult vă doriți să continuați clasa a IX-a la următoarele licee?

Nume liceu	1 – Foarte puțin	2 – Puțin	3 – Mediu	4 – Mult	5 – Foarte mult
...					
...					
...					

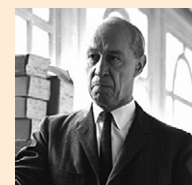
ȘTIAȚI CĂ...?

În activitățile de cercetare statistică în care se utilizează chestionare sunt foarte frecvent întâlnite întrebări la care respondentul (persoana care răspunde) trebuie să bifeze un răspuns din câteva variante date, variantele fiind asociate unor numere secvențiale (scală).

De exemplu: *Cât de mult îți place cartea Harry Potter scrisă de J.K. Rowling?*

1 – Foarte puțin	2 – Puțin	3 – Mediu	4 – Mult	5 – Foarte mult

Această scală de valori pentru răspunsuri poartă numele de *scală Likert*, după numele inventatorului său, psihologul american Rensis Likert.



Rensis Likert



Exersați

1 Care este rezultatul executării secvenței de cod alăturate, dacă la intrare avem $n=6$ și șirul $v=\{1, 7, 3, 17, 27, 5\}$?

- a 1;
- b 2;
- c 3;
- d 0.

```
int v[100], n, i=0, nr=0;
cout<<"n="; cin>>n;
do
{
    cin>>v[i];
    if (v[i]%10==7 && i%2==0) nr++;
    i++;
} while (i<n);
cout<<nr;
```



2 Completați valoarea pe care trebuie să o ia variabila a din secvența de cod alăturată astfel încât să se afișeze rezultatul 24.

```
int v[]={1,2,3,4,5,6}, i=0, a, nr=0;
a=.....;
while (i<6)
{
    if (v[i]>4) {nr++; a++;}
    i++;
}
cout<<nr<<a;
```

3 Se dă un șir de n numere reale, unde $1 \leq n \leq 100$. Să se afișeze dacă acesta conține numere cuprinse în intervalul $[1, 25]$.

Exemplu: **Intrare:** $n=3$ 3.4 72.0 5.2
 Ieșire: Da
 Intrare: $n=3$ 27.1 34.3 76.5
 Ieșire: Nu

4 Afișați numărul elementelor unui șir de n numere întregi, unde $1 \leq n \leq 100$, care sunt divizibile cu numărul 7.

Exemplu: **Intrare:** $n=5$ 13 7 14 3 1
 Ieșire: 2

5 Afișați numărul elementelor unui șir de n numere reale, unde $1 \leq n \leq 100$, care sunt mai mari decât diferența dintre maximul și minimul șirului.

Exemplu: **Intrare:** $n=5$ 7 15 9 6 11
 Ieșire: 2

6 Afișați de câte ori apare într-un șir de n numere reale minimul și maximul său, unde $1 \leq n \leq 100$.

Exemplu: **Intrare:** $n=6$ 1 2 1 7 5 3
 Ieșire: Nr. aparitii minim: 2, nr. aparitii maxim: 1

7 Afișați câte elemente ale unui șir de n numere naturale au 6 divizori diferiți de 1 și de ele însele, unde $1 \leq n \leq 100$.

Exemplu: **Intrare:** $n=4$ 30 1 23 7
 Ieșire: 1

8 Se dă un șir de n numere întregi, unde $1 \leq n \leq 100$. Afișați **Da**, dacă oricare două numere consecutive ale șirului au parități diferite (unul par și celălalt impar), respectiv **Nu**, în caz contrar.

Exemplu: **Intrare:** $n=5$ 2 5 4 3 12
 Ieșire: Da
 Intrare: $n=5$ 2 6 4 3 12
 Ieșire: Nu

9 **Lucru în echipă.** Scrieți în caiete câte o problemă care să se rezolve cu ajutorul unuia dintre algoritmi învățați pentru numărare. Schimbați apoi între voi problemele și rezolvați-le în C++.



Operații matematice bazate pe valoarea elementelor unui șir



Descoperiți

Scrieți în Code::Blocks programele **A**, **B** și **C** propuse mai jos și urmăriți pas cu pas funcționarea lor. În cele ce urmează, veți descoperi noi algoritmi pentru obținerea de rezultate dintr-un șir de valori, calcularea valorilor medii, operații matematice bazate pe valoarea elementelor (sumă/produs), cu aplicabilitate în practică.

A Suma valorilor

Să se calculeze suma valorilor elementelor unui șir cu n numere întregi, unde $1 \leq n \leq 100$.

Aplicabilitate practică: La un concurs școlar, elevii au de rezolvat n subiecte, unde $1 \leq n \leq 100$. La fiecare subiect obțin un punctaj. Să se calculeze suma valorilor punctajelor pe care le-au obținut.

Exemplu: Intrare: $n=5$ 10 15 10 5 10 *Ieșire:* Suma este: 50

Indicații: Pentru a rezolva problema vom aplica un algoritm general pentru calculul sumei mai multor numere.



Rețineți

Suma elementelor unui șir se calculează repetitiv, adunând element cu element. Ea se inițializează cu 0 înainte de a începe adunarea elementelor, pentru ca valoarea sa inițială să nu afecteze valoarea sumei.

Algoritmul pentru calculul sumei elementelor unui șir este următorul:

- 1 Se inițializează suma cu 0.
- 2 Se introduc elementele șirului.
- 3 Se adună la sumă fiecare element în parte, pe măsură ce se introduc.
- 4 Se afișează suma.

Vom stoca șirul de numere într-un tablou unidimensional v cu 100 de elemente, pentru că $1 \leq n \leq 100$. În rest, nu trebuie decât să aplicați algoritmul de mai sus, pas cu pas.

Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{
    int v[100], n, i, s=0; //se initializeaza suma s cu 0 la declarare

    cout<<"n="; cin>>n; //se introduce numarul de elemente

    for (i=0; i<n; i++)
    {
        cin>>v[i]; //se introduc elementele sirului
        s=s+v[i]; //se calculeaza suma elementelor
    }
    cout<<"Suma este:"<<s; //se afiseaza suma

    return 0;
}
```

```
n=5
10 15 10 5 10
Suma este:50
Process returned 0 (0x0) execution time : 10.714 s
Press any key to continue.
```



Aplicați și investigați

Modificați problema de mai sus și calculați produsul valorilor elementelor pare. Investigați care trebuie să fie tipul variabilei care reține produsul, ținând cont de valoarea mare pe care poate să o aibă acesta și de faptul că tipul `int` are ca interval de valori $-2147483648 \dots 2147483647$.



Descoperiți

B Valori medii

Se dă un șir de n numere reale, unde $1 \leq n \leq 31$. Să se afișeze media aritmetică a elementelor șirului și numărul de elemente care au valoarea mai mare decât aceasta.

Aplicabilitate practică: La o stație meteo se fac zilnic măsurători ale temperaturii exterioare într-o lună de vară cu 31 de zile. Realizați un program care să afișeze care este temperatura medie din luna respectivă și numărul de zile în care temperatura a depășit pragul mediu al lunii.

Exemplu: Intrare: $n=6$ 10.5 11 9 15 17 16 Ieșire: Media este: 13.0833 Nr. elemente mai mari decat media: 3

Indicații: Rezolvarea implică utilizarea unui algoritm simplu de calculare a mediei aritmetice, utilizând formula matematică.



Rețineți

Algoritm pentru calculul mediei aritmetice a elementelor unui șir este:

- 1 Se introduc elementele șirului.
- 2 Se calculează suma elementelor pe măsură ce se introduc.
- 3 Se calculează media aritmetică după formula: $media\ aritmetică = \frac{\langle suma\ elementelor \rangle}{\langle nr.\ elementelor \rangle}$.
- 4 Se afișează media aritmetică.



Stocați șirul de numere într-un tablou unidimensional v cu 31 de elemente, pentru că $1 \leq n \leq 31$. În rest, nu trebuie decât să aplicați algoritmul de mai sus, pas cu pas. Suma se calculează conform algoritmului de la problema A.



Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{
    int n, i, nr=0;
    float v[31], m=0;

    cout<<"n="; cin>>n;                //se introduce numarul de elemente
    for (i=0; i<n; i++)
    {
        cin>>v[i];                    //se introduc elementele sirului
        m=m+v[i];                    //se calculeaza suma elementelor
    }
    m=m/n;                            //se calculeaza media aritmetica
    cout<<"Media este:"<<m<<endl;    //se afiseaza media aritmetica

    for (i=0; i<n; i++)                //se calculeaza nr. de elemente mai mari ca media
        if (v[i]>m) nr++;
    cout<<"Nr. elemente mai mari decat media:"<<nr; //se afiseaza nr. de elemente
    return 0;
}
```

```
n=6
10.5 11 9 15 17 16
Media este:13.0833
Nr. elemente mai mari decat media:3
Process returned 0 (0x0)   execution time : 3.510 s
Press any key to continue.
```



Aplicați și investigați

- 1 Investigați posibilitatea de a înregistra valori negative, respectiv pozitive ale temperaturii. Se modifică ceva în program?
- 2 Modificați programul de calcul al mediei aritmetice astfel încât să afișeze media pătratică a elementelor șirului. Media pătratică a elementelor unui șir de numere reale se calculează ca radical din media aritmetică a pătratelor elementelor.

Indicație: În C++, funcția care calculează radicalul este `sqrt()`.



Descoperiți



Rețineți

În C++, sintaxa funcției care calculează radicalul (rădăcina pătrată) este: `sqrt(număr pozitiv)`

Funcția va returna o eroare dacă este aplicată unui număr negativ. Pentru a o putea utiliza, trebuie inclusă biblioteca **cmath**.

```
#include <cmath>
...
int main()
{
    int numar;
    cin>>numar;
    cout<<sqrt(numar);
}
```

c Media armonică

Se dă un șir de n numere reale nenule pozitive, unde $1 \leq n \leq 100$. Să se afișeze media armonică a elementelor șirului. Media armonică a elementelor x_1, x_2, \dots, x_n ale unui șir de n numere reale pozitive este: $m_h = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$

Aplicabilitate practică: Într-o fabrică cu 100 de muncitori, fiecare dintre aceștia realizează o piesă într-un anumit timp (în minute). Calculați timpul mediu de lucru pentru această operație utilizând media armonică.

Exemplu: Intrare: $n=2$ 2 4 **Ieșire:** Media armonică este: 2.66667

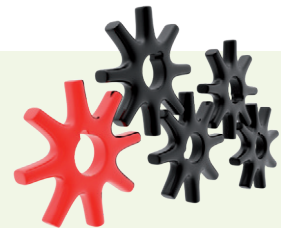
Indicații: Rezolvarea problemei este similară cu cea de la exercițiul B, diferențele constând în aplicarea unei alte formule de calcul a mediei.



Rețineți

Algoritmul pentru calculul mediei armonice a elementelor unui șir este:

- 1 Se introduc elementele șirului.
- 2 Se calculează suma inverselor elementelor pe măsură ce se introduc.
- 3 Se calculează media armonică după formula sa (inversul mediei aritmetice a inverselor numerelor).
- 4 Se afișează media armonică.



Se va utiliza pentru șirul de numere un tablou unidimensional v cu 100 de elemente ($1 \leq n \leq 100$). Apoi se aplică algoritmul.



Programul C++ este:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i;
    float v[100], ma=0;
    cout<<"n="; cin>>n; //se introduce numarul de elemente
    for (i=0; i<n; i++)
    {
        cin>>v[i]; //se introduc elementele sirului
        ma=ma+1/v[i]; //se calculeaza suma elementelor
    }
    ma=n/ma; //se calculeaza media armonica
    cout<<"Media armonica este:"<<ma; //se afiseaza media armonica
    return 0;
}
```

```
n=2
2
4
Media armonica este:2.66667
Process returned 0 (0x0) execution time : 3.674 s
Press any key to continue.
```

ȘTIAȚI CĂ...?

Calculul mediilor are un grad mare de aplicabilitate în analiza statistică. Astfel, în analiza statistică se folosesc mai multe tipuri de medii ca *indicatori* statistici: media aritmetică (m_a), media armonică (m_h), media geometrică (m_g), media pătratică (m_p).



Aplicații și investigații

- Încercați să puneți ca parametru al funcției `sqrt` un număr negativ și vedeți ce se întâmplă.
- Faceți o analiză a aplicării mediei aritmetice și a celei armonice în cazul problemei anterioare. Să presupunem că avem doi muncitori, primul realizează operația respectivă în 10 min, iar al doilea în 15 min. Rulați programele de mai sus (pentru calculul mediei aritmetice și pentru calculul mediei armonice) pentru aceste date și analizați rezultatul în cazul în care cei doi lucrează 60 min.

	Timpul mediu de producere a piesei	Nr. mediu de piese realizate în 60 min de cei doi muncitori
Caz 1: calcul de medie aritmetică	12.5 min	$2 \times 60 \text{ min} / 12.5 \text{ min} = 9.6$
Caz 2: calcul de medie armonică	12 min	$2 \times 60 \text{ min} / 12 \text{ min} = 10$

În mod real, ei produc: $60 \text{ min} / 10 \text{ min} + 60 \text{ min} / 15 \text{ min} = 6 + 4 = 10$ piese. Concluzia este că, în acest caz, calcularea mediei armonice ne duce la o estimare mai exactă a timpului mediu de lucru pentru o piesă.



Exersați

- Să se calculeze suma valorilor elementelor prime ale unui șir de n numere întregi, unde $1 \leq n \leq 100$.
Exemplu: Intrare: n=5 13 23 24 46 52
Ieșire: 36
- Se dă un șir de n numere naturale, unde $1 \leq n \leq 100$. Scrieți un program care calculează suma elementelor șirului a căror ultimă cifră este 6.
Exemplu: Intrare: n=5 12 13 56 3 36
Ieșire: 92
- Să se calculeze produsul celui mai mic element al unui șir de n numere naturale cu cel mai mare element al aceluiași șir, unde $1 \leq n \leq 100$.
Exemplu: Intrare: n=5 12 13 56 3 36
Ieșire: 168
- Un șir de n numere reale, unde $1 \leq n \leq 100$, este stocat într-un tablou unidimensional începând cu poziția 0. Calculați produsul elementelor aflate pe pozițiile pare ale tabloului.
Exemplu: Intrare: n=5 3 4 5 6 7
Ieșire: 105
- În cadrul orei de educație fizică au fost înregistrate rezultatele elevilor la proba de alergare viteză 50 m (în secunde). Faceți un program care să afișeze cel mai mic timp realizat, cel mai mare timp realizat și timpul mediu, utilizând media aritmetică. Numărul total al elevilor este 35.
Exemplu: Intrare: n=5 2.3 5 4.5 2.5 7
Ieșire: Timp minim: 2.3 Timp maxim: 7
Timp mediu: 4.26
- La ora de fizică se realizează un experiment în care fiecare elev al unei clase cu n elevi, unde $1 \leq n \leq 30$, măsoară greutatea unei piese în grame. Calculați greutatea medie a piesei, utilizând media aritmetică, și numărul de elevi care au reușit să o măsoare cu o precizie de mai puțin de 0.2 g.
Exemplu: Intrare: n=5 4.1 4.7 5 5.2 4.3
Ieșire: Greutate medie: 4.66 Nr. elevi: 1
- Se dă un șir de maximum 90 de numere naturale. Să se determine câte dintre elementele acestui șir sunt egale cu produsul vecinilor lor.
Exemplu: Intrare: n=5 1 2 2 6 3 *Ieșire: 2*



Căutarea unui element într-un șir *



Descoperiți

Scrieți în Code::Blocks programele **A** și **B** propuse mai jos și urmăriți pas cu pas funcționarea lor. Veți descoperi noi algoritmi pentru căutarea, într-un șir de valori, a unui element care îndeplinește o anumită condiție.

Căutarea într-un șir de valori poate fi **secvențială** sau **binară**, în funcție de modalitatea de parcurgere a șirului la căutare.

A Căutarea secvențială

Se dă un șir de n elemente numere naturale distincte, unde $1 \leq n \leq 100$. Să se afișeze dacă șirul conține numărul natural X , respectiv poziția sa în șir, considerând că elementele șirului sunt stocate într-un tablou unidimensional începând cu poziția 0.

Exemplu: **Intrare:** $n=4$ **1 34 23 56** $X=23$ **Ieșire:** Da, 2

Indicații: Pentru rezolvarea acestei probleme vom aplica un algoritm în care căutarea se realizează verificând fiecare element în parte, unul după celălalt. Acest tip de căutare este numit **căutare secvențială**.



Rețineți

Algoritmul de căutare secvențială a unei valori într-un șir este următorul:

- 1 Se parcurge șirul de valori de la stânga la dreapta sau de la dreapta la stânga, element cu element.
- 2 Se compară fiecare element în parte cu valoarea dată:
 - a Dacă valoarea elementului este egală cu valoarea căutată, atunci am găsit, iar parcurgerea se încheie.
 - b Altfel, se trece la următorul element și se face verificarea valorii.

În practică, se va considera o variabilă **eGasit** de tip întreg, care poate lua două valori: 1 – dacă elementul a fost găsit în șir, și 0 – dacă nu a fost găsit. Inițial considerăm **eGasit=0** (elementul nu este găsit încă), iar în momentul în care îl găsim, această variabilă își va schimba valoarea în 1.

Se va utiliza pentru șirul de numere un tablou unidimensional a cu 100 de elemente ($1 \leq n \leq 100$). Apoi se aplică algoritmul.



Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{   int a[100], n, i;
    int X, eGasit=0;
    cout<<"n="; cin>>n;           //se introduce numarul de elemente ale sirului
    cout<<"X="; cin>>X;           //se introduce valoarea care se cauta
    for (i=0; i<=n-1; i++)       //se introduc elementele sirului
        cin>>a[i];

    i=0;                          //se incepe cautarea de la elementul cu indice 0
    do                             //se repeta cautarea cat timp nu a fost gasit elementul
        //(!eGasit) si nu am ajuns la capatul sirului cu
        //verificarea (i<=n-1)
    {   if(a[i]==X) eGasit=1;      //daca am gasit elementul, variabila eGasit devine 1
        else i++;                //daca nu am gasit elementul, trecem la elementul urmator
    } while (!eGasit && i<=n-1);
    if (eGasit==1)                //daca am gasit elementul, afisam "Da" si pozitia sa
    {   cout<<"Da"<<endl;
        cout<<i;
    }
    else cout<<"Nu";              //nu am gasit elementul, afisam "Nu"
    return 0;
}
```

```
n=4
X=23
1 34 23 56
Da
2
Process returned 0 (0x0)   execution time : 3.482 s
Press any key to continue.
```

**Aplicați și investigați**

Modificați programul anterior, astfel încât parcurgerea șirului să se facă de la dreapta la stânga. În acest caz, se pleacă în căutare cu o inițializare a variabilei i ($i=n-1$), iar după fiecare verificare cu rezultat negativ, valoarea lui i se decrementează. Condiția de parcurgere a structurii repetitive va fi în acest caz ($!eGasit \ \&\& \ i \geq 0$);

**Descoperiți**

În foarte multe cazuri, în viața reală trebuie să căutăm o informație într-o listă ordonată (de exemplu, un nume într-o listă de rezultate la un examen, un cuvânt într-un dicționar). Pentru a realiza această căutare, ar dura prea mult să verificăm secvențial fiecare nume sau cuvânt în parte. Atât în aceste situații când avem șiruri (liste) de cuvinte ordonate în care trebuie să căutăm, cât și atunci când avem șiruri ordonate de numere, vom aplica un algoritm de căutare care poartă numele de **căutare binară**.

B Căutarea binară

Se dă un șir de n elemente numere naturale distincte ordonate crescător, unde $1 \leq n \leq 100$. Să se afișeze dacă șirul conține numărul natural X și să se afișeze poziția sa în șir, considerând că elementele șirului sunt stocate într-un tablou unidimensional începând cu poziția 0.

Exemplu: **Intrare:** $n=11 \ X=9$ 2 3 6 9 11 15 17 21 23 24 29 **Ieșire:** Da 3

Indicații: În cazul problemei noastre în care șirul de valori este ordonat crescător, pentru căutare se poate aplica optim un algoritm de căutare numit **căutare binară**. Acesta se bazează pe înjumătățiri succesive ale intervalului în care se efectuează căutarea.

**Rețineți**

Algoritmul de căutare binară a unei valori într-un șir este următorul:

Repetăm pașii de mai jos până găsim elementul sau până nu mai avem unde căuta.

Comparăm elementul căutat cu elementul care se află la mijlocul șirului nostru.

- Dacă este egal, atunci am găsit elementul.
- Dacă elementul căutat este mai mic decât mijlocul șirului, atunci șirul în care se va efectua căutarea în continuare se restrânge la jumătatea din stânga, capătul din dreapta pentru subșirul în care vom continua căutarea este elementul imediat anterior mijlocului.
- Dacă elementul căutat este mai mare decât mijlocul șirului, atunci șirul în care se va efectua căutarea în continuare se restrânge la jumătatea din dreapta, capătul din stânga pentru subșirul în care vom continua căutarea devine elementul următor mijlocului.

Caz particular: Considerăm șirul a de numere naturale, cu 10 elemente ordonate crescător, stocate într-un tablou unidimensional a . Căutăm elementul 9. Șirul a în care începem căutarea este:

Elemente:	2	3	6	9	11	15	17	21	23	24	29
------------------	---	---	---	---	----	----	----	----	----	----	----

Indici (poziții): 0 1 2 3 4 5 6 7 8 9 10

Comparăm elementul 9 cu elementul de la jumătatea șirului, $a[5]=15$.

$9 < a[5] \Rightarrow$ căutăm în continuare în prima jumătate a șirului, cea din stânga:

Elemente:	2	3	6	9	11
------------------	---	---	---	---	----

Indici (poziții): 0 1 2 3 4

Comparăm elementul 9 cu elementul de la jumătatea noului șir, $a[2]=6$.

$a[2] < 9 \Rightarrow$ căutăm în continuare în jumătatea dreaptă a șirului:

Elemente:	9	11
------------------	---	----

Indici (poziții): 3 4

Comparăm elementul 9 cu elementul de la jumătatea noului șir, $a[3]=9$.

$a[3] = 9 \Rightarrow$ am găsit elementul, iar poziția sa în șir este 3.



Se va utiliza pentru șirul de numere un tablou unidimensional a cu 100 de elemente ($1 \leq n \leq 100$). Apoi se aplică algoritmul.



Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{   int a[100], n, i;
    int X;
    int eGasit=0;           //variabila ce indica daca am gasit elementul in sir
    int cstg, cdr;         //indicii capetelor stanga/dreapta ale sirului in care se cauta
    int mijloc;           //indicele mijlocului sirului
    cout<<"n="; cin>>n;    //se introduce numarul de elemente ale sirului
    cout<<"X="; cin>>X;    //se introduce valoarea care se cauta
    for (i=0; i<=n-1; i++) //se introduc elementele sirului (ordonate crescator)
        cin>>a[i];
    cstg=0;                //capatul din stg. al sirului in care se cauta este 0
    cdr=n-1;              //capatul din dr. al sirului in care se cauta este n-1
    while (cstg<=cdr && !eGasit) //se repeta cautarea cat timp mai avem unde cauta
        //(cstg<cdr) si nu s-a gasit elementul (!eGasit)
    {   mijloc=(cstg+cdr)/2; //se calculeaza mijlocul sirului

        if (a[mijloc]==X) eGasit=1; //elementul cautat este cel din mijloc

        else //elementul nu este cel cautat

            if (a[mijloc]<X) cstg=mijloc+1; //se va continua cautarea in jumatatea
                //din dreapta

            else cdr=mijloc-1; //se va continua cautarea in jumatatea
                //din stanga

        }
    if (eGasit==1) //daca am gasit elementul, afisam "Da" si pozitia sa
    {   cout<<"Da"<<endl;
        cout<<mijloc;
    }
    else cout<<"Nu"; //nu am gasit elementul, afisam "Nu"
    return 0;
}
```

```
n=11
X=9
2 3 6 9 11 15 17 21 23 24 29
Da
3
Process returned 0 (0x0)   execution time : 11.019 s
Press any key to continue.
```



Aplicații și investigații

- 1 Rulați pas cu pas programul anterior pentru următoarele date de intrare: $n=4$, șirul **12 34 56 78** și elementul care se caută 111. Observați cum anume se iese din structura repetitivă în acest caz.
- 2 Rulați cele două programe **A** și **B** anterior prezentate pe aceleași seturi de date și comparați timpii de execuție. Investigați și explicați rezultatele.



Exersați

- 1 În ce caz de căutare a unui element într-un șir este optim să se aplice algoritmul de căutare binară?
- 2 Scrieți în caiete de câte ori se calculează mijlocul unui șir în cadrul unui algoritm de căutare binară pentru următoarele date de intrare: șirul are 20 de elemente, anume **1 12 14 16 22 56 57 48 89 90 100 101 111 123 134 145 146 157 160 189** și se caută elementul 123.
- 3 Explicați de ce utilizarea unei structuri repetitive cu număr cunoscut de pași, în programul de căutare secvențială, în locul structurii **do ... while** ar duce la o posibilă mărire a timpului de execuție.
Indicație: Gândiți-vă că există cazuri în care chiar primul element este cel căutat.

Sortarea (ordonarea) șirurilor *



Descoperiți

În cele ce urmează, ne propunem să descoperim cum putem realiza sortarea (ordonarea) unui șir de valori stocat într-un tablou unidimensional.



Rețineți

Numim **sortarea unui șir** dispunerea (aranjarea) elementelor șirului într-o anumită ordine (de exemplu, crescătoare sau descrescătoare).

Scrieți în Code::Blocks programele **A** și **B** propuse mai jos și urmăriți pas cu pas funcționarea lor. Aplicați ceea ce ați învățat, experimentați și investigați rezultatele.



A Sortarea prin Metoda bulelor (Bubble sort)

Se dă un șir de n elemente numere naturale, unde $1 \leq n \leq 100$. Să se ordoneze crescător acest șir.

Exemplu: Intrare: $n=6$ 5 3 1 6 Ieșire: 1 3 5 6

Indicații: Vom ordona acest șir utilizând o metodă de sortare numită **Metoda bulelor (Bubble sort)**.



Rețineți

Algoritmul pentru sortarea unui șir prin Metoda bulelor este următorul:

Metoda bulelor constă în **compararea fiecărui element al șirului cu succesorul său**. În cazul în care cele două elemente comparate nu sunt în ordinea potrivită, ele se interschimbă.

Șirul de numere se parcurge de mai multe ori, până când nu mai este nicio interschimbare între elemente.

Pentru a vedea dacă au fost interschimbări între elemente la o parcurgere, se definește o variabilă **eSortat**, care primește valoarea 0 (Fals), dacă există interschimbări, și 1 (Adevărat), dacă nu este nicio interschimbare, caz în care șirul este ordonat corespunzător.

Caz particular: $n=4$, iar șirul este 5 3 1 6. Aplicând algoritmul, obținem:

	Indici	0	1	2	3	
Inițial	Elemente șir	5	3	1	6	
Parcurea 1	$i=0$	3	5	1	6	eSortat=1
	$i=1$	3	1	5	6	eSortat=0
	$i=2$	3	1	5	6	eSortat=0
Parcurea 2	$i=0$	1	3	5	6	eSortat=1
	$i=1$	1	3	5	6	eSortat=0
	$i=2$	1	3	5	6	eSortat=0
Parcurea 3	$i=0$	1	3	5	6	eSortat=1
	$i=1$	1	3	5	6	
	$i=2$	1	3	5	6	

ȘTIAȘI CĂ...?

Această metodă de sortare poartă numele de **Metoda bulelor** sau **Sortare cu bule** deoarece, pe parcursul desfășurării sale, elementele mari sau mici (în funcție de tipul de ordine dorit) înaintază lent spre capătul șirului asemenea unor bule care se ridică la suprafața apei. Acest tip de sortare se mai numește și „sortarea prin propagare“.



Se va utiliza pentru șirul de numere un tablou unidimensional v cu 100 de elemente ($1 \leq n \leq 100$).

Programul C++ este:



```
#include <iostream>
using namespace std;

int main()
{   int n, i, eSortat, m;
    int v[100];
    cout<<"n="; cin>>n;           //se introduce numarul de elemente
    for (i=0; i<=n-1; i++)       //se introduc elementele sirului
        cin>>v[i];

    do                            //se parcurge sirul cat timp inca nu este sortat
    {   eSortat=1;                //pornim cu presupunerea ca sirul este sortat
        for (i=0; i<=n-2; i++)
        {   if(v[i]>v[i+1])       //daca 2 elemente succesive nu sunt in ordinea buna
            {   m=v[i];          //atunci ele se interschimba
                v[i]=v[i+1];
                v[i+1]=m;
                eSortat=0;       //a fost facuta o inversiune, deci sirul nu era sortat
            }
        }
    } while (eSortat==0);
    for(i=0; i<=n-1; i++)       //se afiseaza sirul sortat
        cout<<v[i]<<" ";
    return 0;
}
```

```
n=4
5 3 1 6
1 3 5 6
Process returned 0 (0x0)   execution time : 2.504 s
Press any key to continue.
```



Aplicații și investigații

- 1 Aplicați algoritmul de mai sus pentru a ordona descrescător un șir de numere. Pentru aceasta, trebuie să schimbați condiția de interschimbare a elementelor: `if (v[i]<v[i+1]) {...}`.
- 2 Modificați acest program astfel încât tabloul unidimensional care conține șirul de valori să înceapă de la 1.
- 3 Investigați de ce în programul de mai sus, în cadrul celei de-a doua structuri repetitive `for`, avem expresia de testare `i<=n-2`. Pentru aceasta, uitați-vă la indicii elementelor succesive care sunt comparate.



Descoperiți

B Sortarea prin selecție

Vom rezolva același tip de problemă ca mai înainte, dar folosind o altă metodă de ordonare.

Copiii unei școli au început să economisească bani din bursa lor. Știind că sunt n copii, unde $1 \leq n \leq 100$, ordonați crescător sumele strânse de ei la un moment dat.

Indicații:

Pentru a rezolva această problemă, vom aplica o metodă de sortare numită **sortarea prin selecție**.



Rețineți

Algoritmul pentru sortarea unui șir prin metoda selecției este următorul:

Metoda selecției constă în compararea fiecărui element al șirului cu toate elementele de după el.

În cazul în care cele două elemente comparate nu sunt în ordinea potrivită, ele se interschimbă. După efectuarea tuturor comparațiilor, șirul va fi sortat.

Caz particular: $n=4$, iar șirul este **5 3 1 6**. Aplicând algoritmul, obținem:

Inițial	Indici	0	1	2	3
	Elemente șir	5	3	1	6
$i=0$	$j=1$	3	5	1	6
	$j=2$	1	5	3	6
	$j=3$	1	5	3	6
$i=1$	$j=2$	1	3	5	6
	$j=3$	1	3	5	6
$i=2$	$j=3$	1	3	5	6

Observație: După ce elementul de pe o poziție i este comparat cu toate cele de după el (pozițiile j), pe poziția i va fi plasat cel mai mic dintre elementele comparate.



Se va utiliza pentru șirul de numere un tablou unidimensional v cu 100 de elemente ($1 \leq n \leq 100$).

Programul C++ este:

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, j, m, v[100];
    cout<<"n="; cin>>n;
    for(i=0; i<n-1; i++)
        cin>>v[i];
    for (i=0; i<n-2; i++)
        for (j=i+1; j<n-1; j++)
            if(v[i]>v[j])
            {
                m=v[i];
                v[i]=v[j];
                v[j]=m;
            }
    for(i=0; i<n-1; i++)
        cout<<v[i]<<" ";
    return 0;
}
```

```
n=4
5 3 1 6
1 3 5 6
Process returned 0 (0x0)   execution time : 2.784 s
Press any key to continue.
```



Aplicați și investigați

- Modificați programul de mai sus astfel încât să se ordoneze sumele de bani descrescător. Pentru aceasta, trebuie să schimbați condiția de interschimbare a elementelor: `if (v[i]<v[j]) {...}`.
- Faceți o analiză comparativă a celor două metode de sortare. Analizați numărul de comparații efectuate pentru fiecare în parte.



Exersați

- Ajutați-i pe alpinști să își aleagă vârfurile muntoase pe care să le abordeze. Se știe că vârfurile cele mai periculoase sunt cele mai înalte. Înregistrați înălțimea a n vârfuli, unde $1 \leq n \leq 30$, într-un șir de numere și realizați un program care să le afișeze descrescător pe primele 5 cele mai înalte.
Exemplu: Intrare: $n=9$ 2228 8852 4810 6194 4897 6952 4884 5895 5642 **Ieșire:** 8852 6952 6194 5895 5642
- Se dă un șir de n elemente numere naturale, unde $1 \leq n \leq 100$. Ordonați crescător elementele aflate pe pozițiile pare ale acestui șir, considerând că acesta este stocat într-un tablou unidimensional începând cu poziția 0.
Exemplu: Intrare: $n=6$ 5 1 2 3 7 9 **Ieșire:** 2 5 7
- Se consideră un șir de n numere reale, unde $1 \leq n \leq 100$. Să se ordoneze primele x numere în ordine crescătoare și ultimele numere rămase, în ordine descrescătoare, unde $x < n$.
Exemplu: Intrare: $n=10$ 3 1 5 2 7 3 8 4 6 9 $x=6$ **Ieșire:** 1 2 3 3 5 7 9 8 6 4

Inserarea/ștergerea elementelor unui șir de valori *



Descoperiți

Scrieți în Code::Blocks programele **A**, **B** și **C** propuse mai jos și urmăriți pas cu pas funcționarea lor. Descoperiți algoritmi prezentați pentru inserarea/ștergerea elementelor din tablourile unidimensionale.

A Inserarea unui element într-un tablou unidimensional

Se dă un șir de n numere întregi, stocat într-un tablou unidimensional începând cu poziția 0, unde $3 \leq n \leq 100$. Scrieți un program care să insereze un element cu valoarea 10 pe poziția p , unde $p \leq n-1$.

Exemplu: Intrare: $n=5$ $p=3$ 1 2 3 4 5 *Ieșire:* 1 2 3 10 4 5

Indicații: Programul de față se rezolvă stocând șirul într-un tablou unidimensional și aplicând apoi un algoritm de inserare a unui element nou pe o poziție dată.



Rețineți

Algoritmul de inserare a unui element nou într-un tablou unidimensional pe o poziție p este următorul:

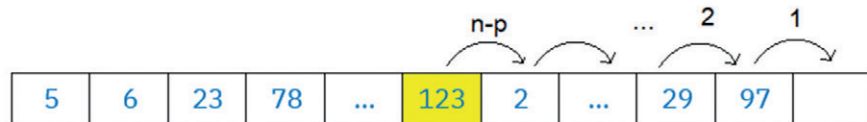
- 1 Se deplasează cu o poziție la dreapta toate elementele începând de la poziția p până la ultima poziție. Dimensiunea logică a tabloului crește cu 1 și se creează un loc pe poziția p , unde se poate insera noul element.
- 2 Se inserează pe poziția p a tabloului noua valoare.
- 3 Se afișează tabloul cu noua sa dimensiune logică $n+1$.

Atenție la declararea tabloului unidimensional. Dimensiunea sa fizică trebuie să fie mai mare sau egală cu dimensiunea logică după inserarea elementului.

Exemplu pentru un șir cu n elemente:

Inițial

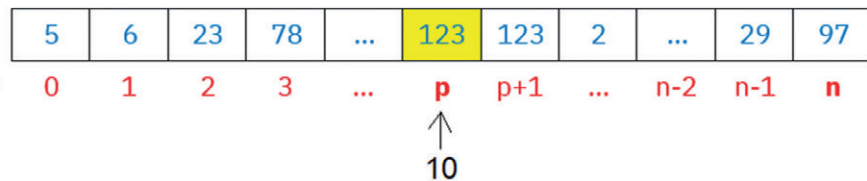
Elementele tabloului:



Indicele (poziția elementelor):

După ce mutăm spre dreapta cu o poziție toate elementele tabloului de pe pozițiile mai mari sau egale cu p

Elementele tabloului:



Indicele (poziția elementelor):

După ce inserăm noua valoare pe poziția p

Elementele tabloului:



Indicele (poziția elementelor):

În cazul nostru, șirul inițial are maximum 100 de elemente, ceea ce înseamnă că după adăugarea noului element va avea dimensiunea de maximum 101 elemente. Astfel, tabloul unidimensional v care va stoca șirul nostru va fi declarat $v[101]$.

Programul C++ este:

```
#include <iostream>
using namespace std;
int main()
{ int v[101], i, n, p;
```

```
n=5
p=3
1 2 3 4 5
1 2 3 10 4 5
Process returned 0 (0x0)   execution time : 6.281 s
Press any key to continue.
```

```

cout<<"n="; cin>>n; //se introduce nr. de elemente
cout<<"p="; cin>>p; //se introduce pozitia pe care inseram element nou

for(i=0; i<=n-1; i++) //se introduc elementele sirului
    cin>>v[i];

for(i=n-1; i>=p; i--) //se deplaseaza elementele cu o pozitie la dreapta
    v[i+1]=v[i];
v[p]=10; //se insereaza noul element pe pozitia p

for(i=0; i<=n; i++) //se afiseaza noul sir
    cout<<v[i]<<' ';

return 0;
}

```



Aplicații și investigații

- 1 Ca un caz particular, aplicați algoritmul de mai sus pentru a insera un element pe poziția 0. Urmăriți pas cu pas execuția programului.
- 2 Explicați ce se întâmplă dacă, în cazul tabloului unidimensional v din problema anterioară, deplasarea la dreapta a elementelor tabloului de la poziția p în sus se face astfel:

```

for(i=p; i<=n-1; i++)
    v[i+1]=v[i];

```



Descoperiți

B Inserare multiplă într-un tablou unidimensional

Inserați, după fiecare element multiplu de 3 al unui șir de 6 numere întregi, un element cu aceeași valoare.

Exemplu: **Intrare:** 1 2 3 4 5 6 **Ieșire:** 1 2 3 3 4 5 6 6

Indicații: Problema se rezolvă utilizând un algoritm de inserare multiplă de elemente într-un tablou unidimensional, derivat din algoritmul de inserare a unui element prezentat la problema A.



Rețineți

Algoritmul de inserare multiplă de elemente într-un tablou unidimensional este următorul:

- 1 Se parcurge tabloul și se identifică, pe rând, fiecare element după care trebuie inserat elementul nou.
- 2 La găsirea acestuia, se aplică algoritmul de inserare de element nou, descris mai sus.
- 3 Dimensiunea logică a tabloului crește cu 1 la fiecare element inserat.

Revenind, deoarece dimensiunea inițială a șirului este 6, în cazul în care toate elementele ar fi multipli de 3, s-ar ajunge la maximum 12 elemente. Astfel, declararea fizică a tabloului unidimensional care stochează șirul de numere este $v[12]$.



Programul C++ este:

```

#include <iostream>
using namespace std;
int main()
{
    int v[12], i, j, l;
    for(i=0; i<=5; i++) //se introduc elementele sirului
        cin>>v[i];

    l=6; //l este dimensiunea logica a sirului, initial este 6
    j=0; //j este indicele elementului pe care-l verificam, incepem cu 0
    while (j<=l-1) //se parcurge sirul (se repeta actiunile cat timp mai sunt elemente)

```

```

1 2 3 4 5 6
1 2 3 3 4 5 6 6
Process returned 0 (0x0)   execution time : 5.600 s
Press any key to continue.

```

```

{   if(v[j]%3==0) //daca elementul este multiplu de 3, urmeaza sa se insereze un altul
        //egal
    {
        //se deplaseaza spre dreapta elementele sirului cu pozitia mai mare
        //ca a elementului gasit
        for(i=l-1; i>=j+1; i--)
            v[i+1]=v[i];
        v[j+1]=v[j]; //se insereaza elementul nou
        l++; //se incrementeaza dimensiunea logica a sirului
        j+=2; //se continua parcurgerea sirului cu urmatorul element,
        //sarindu-l pe cel nou
    }
    else j++; //se continua parcurgerea sirului cu urmatorul element
}
for(i=0; i<=l-1; i++) //se afiseaza sirul rezultat
    cout<<v[i]<<' ';
return 0;
}

```



Aplicați și investigați

Modificați algoritmul de mai sus astfel încât lângă fiecare element să se mai insereze unul. Care va fi dimensiunea logică a tabloului unidimensional v în acest caz? Investigați ce s-ar întâmpla dacă dimensiunea logică a tabloului v ar fi mai mare decât dimensiunea fizică declarată a acestuia.



Descoperiți

c Ștergerea unuia sau a mai multor elemente ale unui tablou unidimensional

Se dă un șir de n numere întregi, stocat într-un tablou unidimensional începând cu poziția 0, unde $3 \leq n \leq 100$. Scrieți un program care să ștergă elementul de pe poziția p , unde $p \leq n-1$.

Exemplu: Intrare: $n=5$ $p=3$ **1 2 3 4 5** Ieșire: **1 2 3 5**

Indicații: Pentru rezolvarea problemei vom aplica un algoritm pentru ștergerea unui element aflat pe o poziție dată într-un tablou unidimensional care stochează șirul.



Rețineți

Algoritmul de ștergere a unui element aflat pe o poziție p într-un tablou unidimensional este următorul:

Se deplasează cu o poziție la stânga toate elementele tabloului, începând de la poziția $p+1$ până la ultima poziție. Astfel se suprascrive elementul de pe poziția p , altfel spus, se elimină. Dimensiunea logică a tabloului scade cu o unitate – ultimul element se ignoră la afișarea șirului nou.

Atenție! În acest caz dimensiunea fizică a tabloului rămâne neschimbată, doar dimensiunea logică se modifică.

Exemplu pentru un șir cu n elemente:

Inițial

Elementele tabloului:

5	6	23	78	...	123	2	34	...	29	97
---	---	----	----	-----	-----	---	----	-----	----	----

Indicele (poziția elementelor): 0 1 2 3 ... p $p+1$ $p+2$... $n-2$ $n-1$

După ce mutăm spre stânga cu o poziție elementele șirului de pe pozițiile mai mari sau egale cu p

Elementele tabloului:

5	6	23	78	...	2	34	...	29	97	97
---	---	----	----	-----	---	----	-----	----	----	---------------

Indicele (poziția elementelor): 0 1 2 3 ... p $p+1$ $p+2$... $n-2$ $n-1$

Programul C++ este:

```
#include <iostream>
using namespace std;
int main()
{   int v[101], i, n, p;
    cout<<"n="; cin>>n;           //se introduce nr de elemente
    cout<<"p="; cin>>p;           //se introduce pozitia elementului care se sterge
    for(i=0; i<=n-1; i++)        //se introduc elementele sirului
        cin>>v[i];
    for(i=p; i<=n-2; i++)        //se muta elementele sirului cu o pozitie la stanga
        v[i]=v[i+1];
    for(i=0; i<=n-2; i++)        //se afiseaza noul sir
        cout<<v[i]<<' ';
    return 0;
}
```

```
n=5
p=3
1 2 3 4 5
1 2 3 5
Process returned 0 (0x0)   execution time : 6.045 s
Press any key to continue.
```

Dacă doriți să ștergeți mai multe elemente ale unui șir, îl stocați într-un tablou unidimensional și apoi aplicați un algoritm de ștergere multiplă a elementelor, derivat din algoritmul anterior.



Rețineți

Algoritmul de ștergere multiplă a elementelor unui tablou unidimensional este următorul:

- 1 Se parcurge tabloul și se identifică, pe rând, fiecare element care trebuie șters.
- 2 La găsirea acestuia, se aplică algoritmul de ștergere a elementului descris anterior.
- 3 Dimensiunea logică a tabloului scade cu 1 la fiecare element șters.



Aplicați și investigați

- 1 Investigați ce s-ar întâmpla dacă în cadrul șirului inițial mutăm cu o poziție la dreapta elementele sale de la poziția $p-1$ la poziția 0, apoi afișăm tabloul unidimensional începând cu poziția 1, ca în secvența de cod alăturată. Este acesta un algoritm alternativ pentru ștergerea elementului p al șirului?


```
for(i=p-1; i>=0; i--)
    v[i+1]=v[i];
for(i=1; i<=n-1; i++)
    cout<<v[i]<<' ';
```
- 2 Aplicați algoritmul de ștergere a unui element al unui tablou în mod repetat, pentru mai multe elemente ale șirului. Urmăriți cum dimensiunea logică a tabloului scade cu 1 la fiecare ștergere.



Exersați

- 1 Care va fi lungimea logică maximă a unui tablou unidimensional de 55 de numere reale ale cărui elemente încep de pe poziția 0, dacă pe toate pozițiile pare se va insera câte un element suplimentar?

a 110 b 83 c 82 d 109
- 2 Scrieți în caiete care dintre următoarele secvențe de cod au ca rezultat ștergerea elementului de pe poziția p al unui tablou unidimensional cu n elemente care încep de la poziția 0:

a `for(i=n-1; i>=p; i--) v[i+1]=v[i];` b `for(i=p+1; i<=n; i++) v[i]=v[i+1];` c `for(i=n-1; i>=p; i--) v[i]=v[i+1];` d `for(i=p; i<=n-2; i++) v[i]=v[i+1];`
- 3 Într-un șir de n numere naturale reprezentat printr-un tablou unidimensional începând de la 0, inserați pe poziția 5 un număr egal cu maximul elementelor șirului, unde $5 \leq n \leq 100$.
Exemplu: Intrare: $n=7$ 1 2 3 4 5 6 7 Ieșire: 1 2 3 4 5 7 6 7
- 4 Avem un șir de n numere naturale, unde $1 \leq n \leq 100$, cu elementele stocate într-un tablou unidimensional începând cu poziția 0. Inserați media aritmetică a elementelor șirului, pe prima poziție din tablou pozitivă și divizibilă cu 3.
Exemplu: Intrare: $n=5$ 1 2 3 4 5 Ieșire: 1 2 3 3 4 5
- 5 Se dă un șir de n numere reale, unde $1 \leq n \leq 100$. Realizați un program care să facă următoarele:
 - a să șteargă elementele care au valoarea egală cu minimul șirului;
 - b să afișeze câte elemente au fost șterse;
 - c să adauge pe prima poziție un element egal cu maximul șirului.



Probleme cu două sau mai multe șiruri



Descoperiți

Scrieți în Code::Blocks programele **A**, **B**, **C** și **D** propuse mai jos și urmăriți pas cu pas funcționarea lor. Descoperiți algoritmi prezentați pentru lucrul cu mai multe șiruri.

A Copierea întocmai a unui șir într-un alt șir

Să se copieze un șir de numere a , care conține n numere reale, într-un șir b de aceeași dimensiune, unde $1 \leq n \leq 100$.

Exemplu: Intrare: $n=5$ șirul a este: **3 4 5 6 7** *Ieșire:* Șirul b este: **3 4 5 6 7**

Indicații: Pentru a rezolva această problemă va trebui să urmați pașii descriși în algoritmul de mai jos.



Rețineți

Algoritmul pentru copierea unui șir de valori într-un alt șir de valori cu aceeași dimensiune este următorul:

- 1 Se citesc elementele șirului inițial care trebuie copiat.
- 2 Se parcurge șirul inițial și se copiază element cu element din primul șir în cel de-al doilea.

Important! Copierea unui vector nu poate fi făcută printr-o atribuire directă de forma $a=b$.

Șirul de numere inițial va fi stocat într-o variabilă a de tip tablou unidimensional, cu dimensiunea 100, acesta fiind numărul maxim posibil de elemente. Șirul b va avea tot același număr de elemente.

Programul C++ este:

```
#include <iostream>
using namespace std;
int main()
{   int a[100], b[100], i, n;

    cout<<"n="; cin>>n;           //se introduce nr de elemente

    cout<<"Introduceti sirul a:"<<endl;
    for (i=0; i<=n-1; i++)      //se citesc elementele sirului a
        cin>>a[i];

    for (i=0; i<=n-1; i++)      //se parcurge sirul initial
        b[i]=a[i];             //se copiaza element cu element in al doilea sir

    cout<<"Sirul b este:"<<endl;
    for (i=0; i<=n-1; i++)      //se afiseaza al doilea sir
        cout<<b[i]<<' ';

    return 0;
}
```

```
n=5
Introduceti sirul a:
1 2 3 4 5
Sirul b este:
1 2 3 4 5
Process returned 0 (0x0)   execution time : 6.235 s
Press any key to continue.
```



Aplicați și investigați

- 1 Compactați primele două structuri repetitive într-una singură, ca în modelul alăturat. Care este avantajul?
- 2 Modificați programul anterior astfel încât cele două șiruri a și b să fie stocate în tablouri unidimensionale începând cu indicele 1.

```
for (i=0; i<=n-1; i++)
{   cin>>a[i];
    b[i]=a[i];
}
```




Descoperiți

B Generarea unui șir nou pe baza unui șir existent – șirurile au aceeași lungime

Se dă un șir de numere a cu n elemente, numere naturale, unde $1 \leq n \leq 100$. Să se construiască un alt șir de numere b , cu n elemente, cu proprietatea că un element al șirului b este dublul elementului aflat pe aceeași poziție din șirul a .

Exemplu: Intrare: $n=5$ șirul a este: **3 4 5 6 7** Ieșire: $b=6$ Șirul b este: **8 10 12 14**

Indicații: Pentru a rezolva această problemă va trebui să urmați pașii descriși în algoritmul de mai jos.



Rețineți

Algoritmul pentru generarea unui șir de valori dintr-un alt șir de valori este următorul:

- 1 Se citesc elementele șirului inițial care trebuie copiat.
- 2 Se parcurge șirul inițial și, în funcție de elementul curent al primului șir, se generează elementul corespunzător al celui de-al doilea șir.

Observație: Al doilea șir poate avea aceeași dimensiune ca primul șir (ca în cazul problemei actuale) sau poate avea o dimensiune diferită, depinzând de algoritmul de generare specificat. Este important ca la declararea șirurilor să se estimeze corect dimensiunea fizică a șirului al doilea.

Programul C++ este:

```
#include <iostream>

using namespace std;

int main()
{
    int a[100], b[100], i, n;

    cout<<"n="; cin>>n;           //se introduce nr de elemente

    cout<<"Introduceti sirul a:"<<endl;
    for (i=0; i<=n-1; i++)       //se citesc elementele sirului a
        cin>>a[i];

    for (i=0; i<=n-1; i++)       //se parcurge sirul initial
        b[i]=2*a[i];           //se calculeaza elementul sirului b

    cout<<"Sirul b este:"<<endl;
    for (i=0; i<=n-1; i++)       //se afiseaza al doilea sir
        cout<<b[i]<<' ';

    return 0;
}
```

```
n=5
Introduceti sirul a:
3 4 5 6 7
Sirul b este:
6 8 10 12 14
Process returned 0 (0x0)   execution time : 3.298 s
Press any key to continue.
```



Aplicați și investigați

- 1 Modificați programul anterior, astfel încât un element al șirului b să fie egal cu pătratul elementului șirului a de pe aceeași poziție ($b[i]=a[i]*a[i];$).
- 2 Realizați un program similar celui de mai sus, cu primele două structuri repetitive **for** compactate ca în activitatea 1 de la programul anterior. Investigați și aici diferența dintre timpul de execuție al programului inițial și cel al programului nou realizat.
- 3 **Lucrați în perechi.** Scrieți în caiete câte două reguli de generare a elementelor unui șir b dintr-un șir a și dați-le colegului de echipă să le implementeze în câte un program C++. Cronometrați timpul în care fiecare dintre voi a realizat programul.



Descoperiți

c Generarea unui șir nou pe baza unui șir existent – șirurile au lungimi diferite

Se dă un șir de numere a cu n elemente, numere naturale, unde $1 \leq n \leq 100$. Considerând că șirul a este stocat într-un tablou unidimensional care începe cu poziția 0, să se construiască un alt șir de numere b , astfel încât elementele acestuia să fie egale cu triplul elementelor aflate pe pozițiile pare ale șirului a .

Exemplu: Intrare: $n=5$ șirul a este: **1 2 3 4 5** Ieșire: Șirul b este: **3 9 15**

Indicații:

Să analizăm dimensiunile celor două tablouri unidimensionale a și b în care se stochează elementele șirurilor cu același nume (a și, respectiv, b).

Caz particular: $n=4$ (par)

Tabloul a este: Elemente

13	22	33	44
----	----	----	----

Indici (poziții) 0 1 2 3

Tabloul b este: Elemente

13	33
----	----

Indici (poziții) 0 1

Caz particular: $n=5$ (impar)

Tabloul a este: Elemente

13	22	33	44	55
----	----	----	----	----

Indici (poziții) 0 1 2 3 4

Tabloul b este: Elemente

13	33	55
----	----	----

Indici (poziții) 0 1 2

Astfel, generalizând pentru cazul problemei noastre:

	Dimensiune fizică	Dimensiune logică
Tabloul a	100	n
Tabloul b	<ul style="list-style-type: none"> 50 dacă n este par 51 dacă n este impar <p><i>Concluzia:</i> Declarația lui b este: $b[51]$</p>	<ul style="list-style-type: none"> câtul lui $n:2$, dacă n este par ($n/2$) (câtul lui $n:2$)+1, dacă n este impar ($n/2+1$)

Se aplică același algoritm ca în cazul problemei anterioare, dar în plus, generarea unui element al șirului b se face doar dacă este verificată condiția ca poziția elementului șirului a să fie pară.

Programul C++ este:

```
#include <iostream>
using namespace std;
int main()
{
    int a[100], b[51], n, i;
    int j=0;                                     //j=pozitia in sirul b va retine la final nr. de
                                                //elemente din b

    cout<<"n="; cin>>n;                          //se citeste numarul de elemente ale sirului

    cout<<"Introduceti sirul a:"<<endl;
    for (i=0; i<=n-1; i++)                      //se citesc elementele sirului a
        cin>>a[i];

    for (i=0; i<=n-1; i++)                      //se parcurge sirul initial
        if (i%2==0)                            //se verifica daca pozitia este para in sirul a
        {
            b[j]=3*a[i];                       //se calculeaza elementul sirului b
            j++;                                //se incrementeaza pozitia elementelor sirului b
        }
    cout<<"Sirul b este:"<<endl;                //aici j contine numarul de elemente ale sirului b
    for (i=0; i<=j-1; i++)                      //se afiseaza al doilea sir
        cout<<b[i]<<' ';

    return 0;
}
```

```
n=5
Introduceti sirul a:
1 2 3 4 5
Sirul b este:
3 9 15
Process returned 0 (0x0)   execution time : 6.204 s
Press any key to continue.
```



Aplicați și investigați

- 1 Modificați programul anterior, astfel încât șirul **b** să aibă elementele egale cu triplul elementelor șirului **a** aflate pe pozițiile pare ale șirului, de la dreapta la stânga. Pentru generarea tabloului unidimensional **b** care stochează elementele șirului, este necesar să parcurgeți tabloul unidimensional **a** de la dreapta la stânga, astfel: `for (i=n-1; i>=0; i--)`.
- 2 Modificați programul anterior, astfel încât șirurile **a** și **b** să fie stocate în tablouri unidimensionale începând de la poziția 1. Pentru aceasta, va trebui să țineti cont că declararea tablourilor care stochează șirurile **a** și **b** va fi: `int a[101], b[51]`.



Descoperiți *

D Combinarea a două șiruri (interclasare)

Se dă un șir **a** cu n elemente și un șir **b** cu m elemente, unde $1 \leq n \leq 100$, $1 \leq m \leq 100$. Ambele șiruri sunt ordonate crescător. Să se construiască un șir **c** care să conțină elementele celor două șiruri în ordine crescătoare.

Exemplu:

Intrare: $n=3$ $m=2$ șirul **a**: 1 3 5 șirul **b**: 2 4

Ieșire: Șirul **c** este: 1 2 3 4 5

Indicații:

Pentru această problemă, ne putem gândi la o primă soluție foarte la îndemână, și anume să copiem elementele celor două tablouri unidimensionale care stochează șirurile **a** și **b** într-un alt tablou unidimensional care stochează șirul **c**, unele după altele, aplicând algoritmul **A** învățat anterior. Apoi, să ordonăm elementele acestui șir **c** și să îl afișăm. Această soluție este ineficientă, ținând cont că elementele șirurilor **a** și **b** sunt deja sortate crescător. Vom aplica un alt algoritm.



Rețineți

Algoritmul pentru combinarea a două șiruri într-unul singur (interclasare) este următorul:

- 1 Parcurgem cele două tablouri **a** și **b** în paralel, cât timp nu s-a terminat de parcurs niciunul dintre șiruri (mai avem elemente de parcurs și într-unul, și în celălalt, în același timp):
 - comparăm la fiecare pas un element din **a** cu un element din **b**. Cel mai mic dintre cele două elemente va fi adăugat în tabloul **c**. Se avansează la următorul element doar în tabloul din care s-a copiat elementul.
- 2 După ce elementele dintr-unul dintre tablouri au fost parcurse în totalitate, vom copia elementele rămase din celălalt tablou în continuare, în tabloul **c**.
Atenție! Tabloul rezultat va avea dimensiunea logică $m+n$ elemente.

Să facem o simulare pentru cele două șiruri de intrare din exemplul oferit:

Tabloul **a** este: Elemente

1	3	5
---	---	---

 Tabloul **b** este: Elemente

2	4
---	---

 Inițializare $n=3$
Indici *i* 0 1 2 Indici *j* 0 1 $m=2$

Pasul 1 (*k* reprezintă indicii tabloului **c**, nou construit)

i	j	k	$i \leq n-1$ & & $j \leq m-1$	comparare elemente	$c[k]$	k	i	j
0	0	0	adevărat	$a[0] < b[0]$	$c[0]=a[0]=1$	1	1	-
1	0	1	adevărat	$a[1] > b[0]$	$c[1]=b[0]=2$	2	-	1
1	1	2	adevărat	$a[1] < b[1]$	$c[2]=a[1]=3$	3	2	-
2	1	3	adevărat	$a[2] > b[1]$	$c[3]=b[1]=4$	4	-	2
2	2	4	fals					

Pasul 2

i	k	$i \leq n-1$	$c[k]$	k	i
2	4	adevărat	$c[4]=a[2]=5$	5	3
3	5	fals			

Elementele celor două șiruri a și b sunt stocate în tablouri unidimensionale de dimensiunea fizică 100 și dimensiuni logice n, respectiv m. Se aplică apoi algoritmul descris anterior. Tabloul c va avea dimensiunea fizică 200 și dimensiunea logică m+n.



Programul C++ este:

```
#include <iostream>
using namespace std;

int main()
{
    int a[100], b[100], n, m, i=0, j=0;
    int c[200], k=0;           //se declara tabloul unidimensional c, initial gol(k=0)

    cout<<"n="; cin>>n;
    cout<<"m="; cin>>m;

    cout<<"Introduceti sirul a:"<<endl;
    for (i=0; i<=n-1; i++)     //se citesc elementele sirului a
        cin>>a[i];

    cout<<"Introduceti sirul b:"<<endl;
    for (i=0; i<=m-1; i++)     //se citesc elementele sirului b
        cin>>b[i];
    i=0;
    while (i<=n-1 && j<=m-1)   //cat timp mai sunt elemente in ambele tablouri a si b
                                //concomitent
    {   if (a[i]<b[j])           //daca elementul curent al sirului a < elementul curent al
                                    //sirului b
        {   c[k]=a[i];           //se copiaza in sirul c elementul mai mic, cel din sirul a
            k++;                 //se incrementeaza pozitia in sirul c
            i++;                 //se inainteaza in sirul a
        }
        else                   //daca elementul curent al sirului b <= elementul curent al
                                //sirului a
        {   c[k]=b[j];           //se copiaza in sirul c elementul mai mic, cel din sirul b
            k++;                 //se incrementeaza pozitia in sirul c
            j++;                 //se inainteaza in sirul b
        }
    }

    while(i<=n-1)              //se copiaza ce a mai ramas din sirul a, daca a mai ramas
    {   c[k]=a[i];
        k++;
        i++;
    }

    while (j<=m-1)             //se copiaza ce a mai ramas din sirul b, daca a mai ramas
    {   c[k]=b[j];
        k++;
        j++;
    }

    cout<<"Sirul c este:"<<endl;
    for (i=0; i<=m+n-1; i++)   //se afiseaza al treilea sir
        cout<<c[i]<<' ';
    return 0;
}
```

```
n=3
m=2
Introduceti sirul a:
1 3 5
Introduceti sirul b:
2 4
Sirul c este:
1 2 3 4 5
Process returned 0 (0x0)   execution time : 6.242 s
Press any key to continue.
```





Aplicați și investigați *

- Rezolvați problema folosind algoritmul descris la începutul indicațiilor ca fiind ineficient, apoi comparați timpul de execuție al acestuia cu timpul de execuție al programului rezolvat prin algoritmul de interclasare. Observați diferența de timp și gândiți-vă la numărul de comparații pe care le efectuează fiecare algoritm în parte.
- Amintiți-vă faptul că secvența de două instrucțiuni `c[k]=b[j]; k++;` este echivalentă cu `c[k++]=b[j];`. Transformați codul programului astfel încât să conțină secvențe comprimate de acest tip.



Exersați



- Care este rezultatul afișat în urma rulării secvenței de cod alăturate dacă șirul `a` este inițializat cu `{7, 4, 6, 67, 31, 9, 17, 22}`?

a 16 484 b 49 900 c 36 81 d 49 36 900

- Se dă un șir de numere `a` cu `n` elemente, numere naturale, unde $1 \leq n \leq 100$. Să se construiască un alt șir de numere `b`, care conține elementele primului șir, în ordine inversă.

Exemplu: Intrare: `n=5 a: 2 3 4 5 6` Ieșire: `b: 6 5 4 3 2`

- Se dă un șir de numere `a` cu `n` elemente, numere naturale, unde $1 \leq n \leq 100$. Să se construiască un alt șir de numere `b`, cu `n` elemente, cu proprietatea că un element al șirului `b` este înșesitul elementului aflat pe aceeași poziție din șirul `a`.

Exemplu: Intrare: `n=5 a: 2 3 4 5 6` Ieșire: `b: 12 18 24 30 36`

- Se dă un șir de numere `a` cu `n` elemente, numere naturale nenule, unde $1 \leq n \leq 100$. Să se construiască un alt șir de numere `b`, cu `n` elemente, cu proprietatea că un element al șirului `b` este egal cu restul împărțirii elementului șirului `a`, aflat pe aceeași poziție, la suma cifrelor sale.

Exemplu: Intrare: `n=3 a: 12 31 44` Ieșire: `b: 0 3 4`

- Scrieți un program care să copieze elementele pare ale unui șir cu `n` elemente numere naturale, unde $1 \leq n \leq 100$, într-un alt șir. Afișați câte dintre elementele celui de-al doilea șir sunt numere prime.

Exemplu: Intrare: `n=5 a: 2 3 4 5 6` Ieșire: `b: 2 4 6` nr. prime: 1

- Scrieți un program care să copieze elementele pare ale unui șir cu `n` elemente numere naturale, unde $1 \leq n \leq 100$, într-un alt șir, în ordine inversă.

Exemplu: Intrare: `n=5 a: 2 3 4 5 6` Ieșire: `b: 6 4 2`

- Se dă un șir de numere `a` cu `n` elemente, numere naturale, unde $1 \leq n \leq 100$. Considerând că șirul `a` este stocat într-un tablou unidimensional începând cu poziția 0, să se construiască un șir de numere `b`, astfel încât pentru fiecare element al lui `a` să se genereze în șirul `b` două elemente, unul egal cu dublul acestuia și unul egal cu triplul său.

Exemplu: Intrare: `n=4 a: 1 2 3 4` Ieșire: `b: 2 3 4 6 6 9 8 12`

- Se dau două șiruri de numere, `a` și `b`, cu `n` elemente, numere naturale, unde $1 \leq n \leq 100$. Să se construiască un șir de numere, `c`, cu proprietatea că fiecare element al său este suma elementelor de pe aceleași poziții ale lui `a` și `b`.

Exemplu: Intrare: `n=4 a: 1 2 3 4 b: 5 6 7 8` Ieșire: `c: 6 8 10 12`

- Utilizați algoritmul de interclasare pentru a rezolva problema anterioară **D**, dar considerând șirurile `a` și `b` ordonate descrescător. Șirul `c` se cere a fi construit tot descrescător.

Exemplu: Intrare: `n=3 m=2 a: 5 3 1 b: 4 2` Ieșire: `c: 5 4 3 2 1`

```
for (i=0; i<=n-1; i++)
    cin>>a[i];
j=0;
for (i=0; i<=n-1; i++)
    if (a[i]%2==0 && i%3==1)
    { b[j]=a[i]*a[i];
      j++;
    }
for (i=0; i<=j-1; i++)
    cout<<b[i]<<' ';
```



PROIECT – Utilizarea șirurilor de numere în robotică



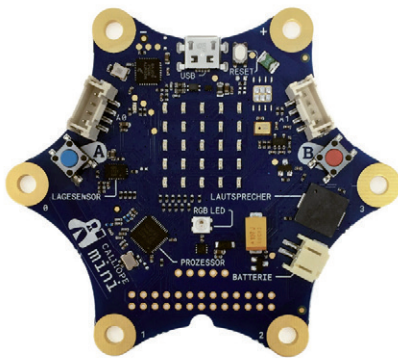
Descoperiți

Șirurile de numere se pot folosi în multe domenii, inclusiv în programarea unui robot virtual.

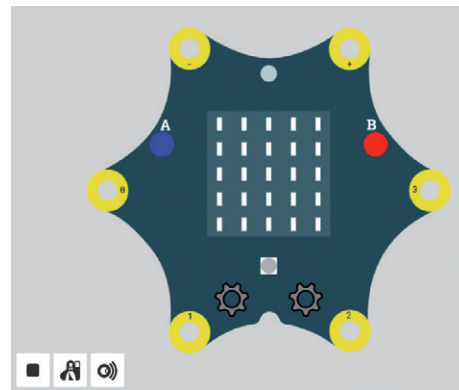
Vom folosi robotul didactic **Calliope mini**, pe care îl vom programa în mediul de lucru Open Roberta (<https://lab.open-roberta.org/>). Prin urmare, vom folosi robotul virtual în fereastra de simulare (accesabilă de la butonul **SIM**).

Programul Robotul muzicant

Faceți un program care, pentru un număr de 10 repetiții, citește ce butoane sunt apășate (**pin 1** sau **pin 2**) dintre butoanele lui **Calliope mini** și pune într-o listă „1” pentru **pin 1** apăsat și „-1” pentru **pin 2** apăsat. Apoi, **Calliope mini** parcurge lista creată și pentru fiecare „1” cântă o notă, iar pentru fiecare „-1” cântă altă notă.



Calliope mini, format fizic



Calliope mini, format virtual, în simulatorul din Open Roberta



Modalitate de rezolvare:

Parcurgeți următoarele etape: creați lista în care se va memora șirul de cifre „1” și „-1”; pentru un număr de 10 iterații, citiți butoanele apășate de pe robot (**pin 1** sau **pin 2**) și inserați în listă numărul corespunzător („1” sau „-1”); parcurgeți lista astfel formată și puneți robotul să cânte o anumită notă pentru „1” și o altă notă pentru „-1” (de preferat să alegeți note distanțate în gamă pentru a se auzi bine diferența dintre ele).

În rezolvarea alăturată, pentru ultimul pas s-a ales soluția în care se parcurge șirul, se ia primul element din șir și se analizează: dacă este „1”, se cântă o notă, altfel e evident că este „-1” și se cântă cealaltă notă, apoi este eliminat din șir; procesul se reia până se termină numerele din șir.

Sunetele alese sunt scurte (optimi) pentru un feedback rapid al programului.

Pentru a se evidenția fiecare apășare de senzor de pe robot (**pin 1** sau **pin 2**), am introdus și afișarea unei imagini formate din leduri aprinse, pentru 100 ms.

Programul realizat în Open Roberta este cel din imaginea alăturată.

```

+ start
- variable SIR : List Number ← + create empty list : Number
repeat 10 times
do
  wait ms 100
  + wait until pin 1 pressed? or pin 2 pressed?
  + if pin 1 pressed?
  do
    in list SIR insert at last as 1
    show image [LED pattern]
    wait ms 100
    clear display
  + if pin 2 pressed?
  do
    in list SIR insert at last as -1
    show image [LED pattern]
    wait ms 100
    clear display
  wait ms 1000
  show image [LED pattern]
repeat length of SIR times
do
  wait ms 100
  + if in list SIR get and remove first = 1
  do
    play eighth note fis/ges
  else
    play eighth note ais"/b"
  
```



Să vedem acum și programul în C++ care face acest lucru (program generat automat de Open Roberta):

```

1. #define _GNU_SOURCE
2. #include "MicroBit.h"
3. #include "NEPODefs.h"
4. #include <list>
5. #include <array>
6. #include <stdlib.h>
7. MicroBit _uBit;
8.
9.
10. std::list<double> __SIR;
11.
12. int main()
13. {
14.     _uBit.init();
15.     __SIR = {};
16.
17.     for (int __k0 = 0; __k0 < 10; __k0 += 1) {
18.         _uBit.sleep(100);
19.         while (true) {
20.             if ( _uBit.io.P0.isTouched() || _uBit.io.P1.isTouched() ) {
21.                 break;
22.             }
23.             _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
24.         }
25.         if ( _uBit.io.P0.isTouched() ) {
26.             __SIR.push_back(1);
27.             _uBit.display.print(MicroBitImage("0,0,0,0,0\n0,0,0,0,0\n0,0,0,0,0\n255,0,0,0,255\n0,255,255,255,0\n"));
28.             _uBit.sleep(100);
29.             _uBit.display.clear();
30.         }
31.         if ( _uBit.io.P1.isTouched() ) {
32.             __SIR.push_back(-1);
33.             _uBit.display.print(MicroBitImage("0,0,0,0,0\n0,0,0,0,0\n0,0,0,0,0\n255,0,0,0,255\n0,255,255,255,0\n"));
34.             _uBit.sleep(100);
35.             _uBit.display.clear();
36.         }
37.         _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
38.     }
39.     _uBit.sleep(1000);
40.     _uBit.display.print(MicroBitImage("0,255,0,255,0\n255,255,255,255,255\n255,255,255,255,255\n0,255,255,255,0\n0,255,0,255,0\n"));
41.     for (int __k1 = 0; __k1 < ((int) __SIR.size()); __k1 += 1) {
42.         _uBit.sleep(100);
43.         if ( _getAndRemoveListElementByIndex(__SIR, 0) == 1 ) {
44.             _uBit.soundmotor.soundOn(184.997); _uBit.sleep(250); _uBit.soundmotor.soundOff();
45.         } else {
46.             _uBit.soundmotor.soundOn(932.328); _uBit.sleep(250); _uBit.soundmotor.soundOff();
47.         }
48.         _uBit.sleep(_ITERATION_SLEEP_TIMEOUT);
49.     }
50.     release_fiber();
51. }
52.

```



Exersați

- 1 Lucru în echipă.** Realizați, lucrând în echipe, programul care să rezolve următoarea problemă:
Robotul Calliope mini creează o listă cu numere aleatoare întregi, între -100 și 100. Alegeți acțiuni diferite pentru numerele negative și pentru cele pozitive, apoi rulați programul.
- 2 Portofoliu.** Realizați un program care să rezolve problema de mai jos:
Un robot trebuie să traverseze un teren dreptunghiular presărat cu obstacole. Dacă întâlnește un obstacol, trebuie să îl ocolească. Robotul primește șirul de numere: {0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0}. Înțelegem astfel: „0” înseamnă că în fața robotului nu este obstacol, iar „1” înseamnă că este obstacol. Înainte de a încerca să înainteze, robotul analizează lista pentru a vedea dacă în fața lui are sau nu obstacol. Dacă nu are, înaintează 10 cm, iar dacă are obstacol, se mută 10 cm la dreapta și se repoziționează cu fața spre direcția inițială. Repetă astfel analiza și mișcarea, până parcurge tot șirul de numere. Câți centimetri a parcurs robotul pentru a traversa terenul?



Recapitulare

Rezolvați în caiete sau pe calculator.

- 1 Se consideră următorul tablou unidimensional $a=\{2, 7, 1, 9, 4, 5, 8\}$ și două variabile întregi i și j . Dacă primul element al tabloului a are indicele 0, atunci rularea secvenței de cod alăturate va modifica elementele tabloului a astfel:

- a** $a=\{7, 2, 9, 1, 5, 4, 8\}$;
b $a=\{7, 1, 9, 4, 5, 8, 2\}$;
c $a=\{5, -6, 8, -5, 1, 3, 8\}$;
d $a=\{7, 2, 9, 1, 5, 8, 4\}$.

Indicație: Secvența de cod $a[i]=a[i]+a[j]$; $a[j]=a[i]-a[j]$; $a[i]=a[i]-a[j]$; realizează interschimbarea valorii lui $a[i]$ cu $a[j]$.

```
i=0;
while(i<=5)
{
  j=i+1;
  a[i]=a[i]+a[j];
  a[j]=a[i]-a[j];
  a[i]=a[i]-a[j];
  i=i+1;
}
```

- 2 Un tablou unidimensional a este format din n numere întregi, unde $1 \leq n \leq 100$. Primul element începe de la 1. Scrieți în caiete corespondențele dintre secvențele de cod din prima coloană și descrierea rezultatelor la rularea lor din a doua coloană.

```
a. for(i=n-1; i>=1; i--)
    if (a[i]<a[i+1])
    {
        x=a[i];
        a[i]=a[i+1];
        a[i+1]=x;
    }
```

1. Se înlocuiește primul element cu cel mai mic din șir.

```
b. for(i=1; i<n; i++)
    if (a[i]>a[i+1])
    {
        x=a[i];
        a[i]=a[i+1];
        a[i+1]=x;
    }
```

2. Se înlocuiește primul element cu cel mai mare din șir.

```
c. for(i=n-1; i>=1; i--)
    if(a[i]>a[i+1])
    {
        x=a[i];
        a[i]=a[i+1];
        a[i+1]=x;
    }
```

3. Se înlocuiește ultimul element cu cel mai mare din șir.

Indicație: Priviți cu atenție exemplul alăturat în care este prezentată execuția secvenței de cod 1 pentru șirul $a=\{3, 5, 1, 2, 9\}$ și $n=5$. Se observă că are corespondent descrierea rezultatului la rulare **b**.

	a[1]	a[2]	a[3]	a[4]	a[5]
i	3	5	1	2	9
4	3	5	1	9	2
3	3	5	9	1	2
2	3	9	5	1	2
1	9	3	5	1	2

- 3 La o anumită disciplină mediile elevilor din școală sunt înregistrate și studiate în vederea evaluării procesului educațional. Afișați numărul elevilor care au media maximă 10, al elevilor cu mediile de 8 și 9, al celor cu mediile 6 și 7, al celor cu mediile de 5 și 6 și al celor cu mediile mai mici de 5. Numărul elevilor din școală este n , unde $1 \leq n \leq 100$. *Indicație:* Se va utiliza algoritmul Statistică de la lecția „Algoritmi de numărare“.

- 4* Se dă un șir a format din n numere naturale, unde $1 \leq n \leq 100$. Ordonăți crescător acest șir și afișați-l. Apoi, generați și afișați un al doilea șir, b , astfel încât un element de pe poziția X al lui b să fie egal cu suma dintre elementul X al lui a și elementul de pe poziția următoare. Construiți apoi un șir c compus din elementele celor două șiruri a și b , ordonat, de asemenea, crescător și afișați-l. Afișați dacă numărul 10 aparține șirului c .

Indicație: Ordonarea crescătoare a șirului a se va face utilizând unul dintre algoritmi de la lecția „Sortarea (ordonarea) șirurilor“. Apoi, se va genera șirul b care va conține $n-1$ elemente. El este, de asemenea, ordonat crescător. Pentru generarea șirului c se poate utiliza algoritmul de interclasare de la lecția „Probleme cu două sau mai multe șiruri“, iar pentru a căuta elementul cu valoarea 10 în șir se va aplica unul dintre algoritmi de căutare din lecția „Căutarea unui element într-un șir“.

Autoevaluare

Rezolvați în caiete sau pe calculator.

- 1 Șirul X conține elementele: 2, 4, 6, 8, 10, 12, ... Care este valoarea elementului de pe poziția 35 din șir?
 a 72; b 70; c 700; d 35.

Indicație: Găsiți regula după care se formează elementele șirului.

- 2 Dacă într-un tablou unidimensional n dorim să stocăm 4 note ale unui elev la o materie și media acestora, care este declararea corectă dintre cele de mai jos?
 a float $n[4]$; b int $n[4]$; c float $n[5]$; d int $n[5]$.

Indicație: Gândiți-vă la tipul și numărul datelor care se stochează.

- 3 Tabloul unidimensional v conține elementele: 1, 2, 3, 4, 5, 6, 7, 8, 9, începând cu indicele 0. Completați valoarea lipsă din secvența alăturată astfel încât la rularea ei să se afișeze 357911.

```
a=1;
for (i=0; i<=...; i++)
cout<<v[i]*2+a;
```

Indicație: Calculați valoarea care se afișează la fiecare pas, valoare care compune rezultatul afișat.

- 4 Se introduce de la tastatură un șir de n numere naturale, unde $1 \leq n \leq 100$. Scrieți un program care afișează elementele șirului în următoarea ordine: ultimul, primul, penultimul, al doilea etc.

Exemplu: **Intrare:** $n=5$ 1 2 3 4 5 **Ieșire:** 5 1 4 2 3

Indicație: Analizați și găsiți regula de parcurgere a indicilor elementelor.

- 5 Ana știe care a fost totalul veniturilor ei din fiecare lună a unui an. Scrieți un program care afișează luna în care ea a avut cel mai mare venit.

Exemplu: **Intrare:** 4234 4432 4222 4001 4978 4987
5223 4455 5500 4378 3226 3190

Ieșire: Luna septembrie

Indicație: Problema inițială se transpune într-una de aflare a maximumului elementului unui șir de valori.



- 6 Se dă un șir de n numere întregi, unde $1 \leq n \leq 100$. Să se verifice dacă toate elementele șirului au ultima cifră egală cu 3.

Exemplu: **Intrare:** $n=4$ 3 12 13 23 **Ieșire:** Nu; **Intrare:** $n=4$ 3 13 133 23 **Ieșire:** Da

Indicație: Este o problemă de verificare a existenței elementelor care au anumite proprietăți. Amintiți-vă de problemele cu numere, în care lucrați cu cifrele acestora.

- 7* Se consideră un șir de n numere reale distincte, mai mici decât 1 000 000, unde $1 \leq n \leq 100$. Să se ordoneze numerele aflate înaintea minimumului șirului în ordine descrescătoare și ultimele numere rămase, în ordine crescătoare. Numărul minim rămâne pe loc.

Exemplu: **Intrare:** $n=6$ 3 2 5.2 1 7 3 **Ieșire:** 5.2 3 2 1 3 7

Indicație: Se va găsi minimumul șirului, apoi se va aplica oricare dintre algoritmi de sortare învățați pentru fiecare dintre cele două subșiruri (cel de dinaintea minimumului și cel de după minim).

Punctaje:

1 – 10 puncte; 5 – 20 puncte;
 2 – 5 puncte; 6 – 10 puncte;
 3 – 15 puncte; 7 – 20 puncte.
 4 – 20 puncte;

Punctaj maxim: 100 puncte.
 Timp de lucru: 50 minute.

Răspunsurile se găsesc
 la finalul manualului.

Insuficient: 0-50 p
 Suficient: 51-60 p
 Bine: 61-80 p
 Foarte bine: 81-90 p
 Excelent: 91-100 p

Recapitulare finală

Rezolvați în caiete sau pe calculator.

Alegeți răspunsurile corecte și răspundeți cerințelor următoare:

- 1 Într-o foaie de calcul tabelar Excel, celula G11 se află:
- la intersecția dintre rândul G și coloana 11;
 - la intersecția dintre rândul 11 și coloana G;
 - la intersecția dintre rândul G1 și coloana 1;
 - la intersecția dintre rândul G11 și coloana 1.

- 2 Ce valoare se va afișa în celula D5 dacă într-o foaie de calcul tabelar Excel introducem valorile la fel ca în imaginea de mai jos, iar în celula D5 se introduce formula =IF(B5<100,100,200)-IF(C5>20,20,10)?

- 90;
- 190;
- 80;
- 180.

	A	B	C	D
5	1	75	21	
6	2	150	15	



- 3 Marius face trei măsurători ale lungimii pentru trei piese și înscrie rezultatele într-o foaie de calcul tabelar, calculând minimul și maximum pentru valorile înscrise pe fiecare rând în parte. Precizați cărui rând îi corespunde fiecare dintre următoarele formule:

• =MIN(B2,C2,D2)

- 2;
- 3;
- 4;
- niciunul.

• =MAX(B4,C4,D4)

- 2;
- 3;
- 4;
- niciunul.

	A	B	C	D	E	F
1		I1	I2	I3	Min	Max
2	piesa 1	12.50	12.75	12.60		
3	piesa 2	45.00	45.56	44.97		
4	piesa 3	23.00	23.25	22.54		

- 4 Paul a avut în primul semestru din clasa a V-a media generală 8,60. El și-a mărit media cu 0,20 în fiecare semestru, până la finalul clasei a VIII-a. Realizați un tabel care arată mediile semestriale ale lui Paul în toate cele 8 semestre din clasele V-VIII și reprezentați grafic evoluția mediei semestriale a lui Paul.

- 5 Zona de celule A4:E4 include următoarele celule:

- A4, E4;
- A4, C4, E4;
- A4, B4, C4, D4, E4;
- B4, C4, D4.

- 6 Ce este o subpagină web?

- 7 Odată activată opțiunea de partajare și colaborare pe Google Sites pentru un coleg (la un proiect X), se mai poate opri partajarea pentru a bloca accesul colegului la respectivul proiect?

- 8 Alegeți A dacă enunțul este adevărat și F dacă enunțul este fals:

- Un carusel de imagini nu poate avea mai multe pagini. A / F
- În Google Sites, la pagina Acasă nu poate fi adăugată o subpagină. A / F
- Pe o pagină din Google Sites putem include filme YouTube, diagrame Excel, imagini. A / F

- 9 Asociați imaginea butonului cu acțiunea care-i corespunde în Google Sites:

a.		1.	Inserarea unei hărți în pagină
b.		2.	Partajarea accesului la proiect
c.		3.	Previzualizarea paginii

10 Cum putem publica un site realizat cu Google Sites?

- a pe un domeniu propriu de tipul www.domeniulmeu.ro;
- b la o adresă pusă la dispoziție de Google, de tipul https://sites.google.com/view/siteulmeu;
- c niciuna dintre acestea; pentru publicare avem nevoie de un server propriu.

11 Elementul al 10-lea al unui tablou unidimensional a ale cărui elemente sunt stocate începând cu indicele 0 se accesează în C++ astfel:

- A[10];
- a[9];
- a[10];
- A[9];
- a(9).

12 Ce afișează secvența de instrucțiuni alăturată?

- a 1454723
- b 326645
- c 1345247662345
- d 2347451

```
int a[]={1,3,45,2,47,66,23,45}, i;
for(i=0; i<=7; i=i+2)
    cout<<a[i];
```

13 Ce elemente va avea șirul de numere a după executarea secvenței de cod alăturate, dacă la intrare el conține elementele {1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6}?

- a {6, 6, 2, 2, 16, 16, 4, 4, 26, 26, 6, 6}
- b {1, 1, 11, 11, 3, 3, 21, 21, 5, 5, 31, 31}
- c {1, 1, 2, 2, 16, 16, 4, 4, 5, 5, 31, 31}
- d {1, 1, 2, 2, 3, 3, 21, 21, 5, 5, 6, 6}

```
for(i=11; i>=0; i--)
    if (a[i]%2==0)
        a[i]=a[i]*5+1;
```

14 Completați valoarea inițială lipsă a variabilei S astfel încât secvența de cod alăturată să afișeze media aritmetică a elementelor divizibile cu 3 ale unui tablou unidimensional a.

```
int a[10], n, i, nr=0;
float S;
S=...;
cout<<"n=";cin>>n;
for(i=0; i<=n-1; i++)
{
    cin>>a[i];
    if(a[i]%3==0)
        {S=S+a[i]; nr++;}
}
S=S/nr;
cout<<S;
```

15 * Se consideră un tablou unidimensional a cu n elemente întregi, unde $1 \leq n \leq 100$. În secvența de cod alăturată se realizează:

- a căutarea binară a elementului cu valoarea X;
- b căutarea secvențială a elementului cu valoarea X;
- c căutarea secvențială a elementelor care nu au valoarea X;
- d căutarea binară a elementelor care nu au valoarea X.

```
int a[100], n, i;
int X, eGasit=0;
...
for(i=0; i<=n-1; i++)
{
    if(a[i]==X) eGasit=1;
    i++;
}
if (eGasit==1) cout<<"Da";
else cout<<"Nu";
```

16 În secvența de cod alăturată, câte elemente ale tabloului unidimensional a memorează valori care au cifra unităților egală cu 8?

- a 5;
- b 6;
- c 8;
- d 0.

```
for(i=0; i<15; i++)
    a[i]=3*i+i*i;
```

17 Se consideră un tablou unidimensional v cu n elemente întregi, unde $1 \leq n \leq 100$. Alegeți varianta corectă pentru ca secvența de instrucțiuni de alături să sorteze elementele tabloului în ordine crescătoare.

- a for(j=n-1; j>=i-1; j--)
- b for(j=n; j>=i+2; j--)
- c for(j=n-1; j>=i+1; j--)

```
for(i=0; i<=n-2; i++)
    for(...;...; j--)
        if(v[i]>v[j])
        {
            m=v[i];
            v[i]=v[j];
            v[j]=m;
        }
```

18 * Se consideră un tablou unidimensional a cu n elemente, unde $1 \leq n \leq 100$, în care primul element este memorat pe poziția 0. Secvența de cod alăturată realizează:

- a adăugarea unui element pe poziția k ;
- b eliminarea elementului aflat pe poziția k ;
- c eliminarea elementului aflat pe poziția $k-1$;
- d eliminarea elementului aflat pe poziția $k+1$.

```
int k, i, a[100], n;
...
for(i=k+1; i<=n-2; i++)
    a[i]=a[i+1];
```

19 Secvența de cod alăturată realizează:

- a copierea elementelor tabloului unidimensional a în b ;
- b copierea elementelor tabloului unidimensional b în a ;
- c copierea elementelor tabloului unidimensional a în b , în ordine inversă;
- d copierea elementelor tabloului unidimensional b în a , în ordine inversă.

```
int a[10], b[10], n, i;

cout<<"n=";cin>>n;
for(i=0; i<=n-1; i++)
    cin>>a[i];
for(i=n-1; i>=0; i--)
    b[i]=a[i];
```

Pentru fiecare dintre exercițiile de mai jos, scrieți programele corespunzătoare și testați-le pe calculator.

20 Se consideră un șir de n numere naturale, unde $1 \leq n \leq 100$. Scrieți un program care:

- a afișează elementele șirului din 2 în 2, de la stânga la dreapta;
- b afișează elementele șirului cu indici pari, de la dreapta la stânga;
- c afișează câte elemente divizibile cu 33 sunt în șir;
- d afișează suma elementelor multipli de 3 din șir;
- e afișează numărul elementelor cuprinse între media aritmetică a elementelor din șir și maximul șirului;
- f afișează dacă există în șir numărul 77.

Exemplu: Dacă $n=7$, iar șirul este **11 22 33 44 55 66 77**, atunci se afișează: **a 11 33 55 77;** **b 77 55 33 11;**
c Nr. elem. divizibile cu 33=2; **d** Suma elem. multipli de 3=99; **e** Nr. elemente=4; **f** Exista.

21 Se dă un șir de numere naturale de n elemente, unde $1 \leq n \leq 100$. Scrieți un program care să afișeze câte dintre elementele acestui șir au aceeași sumă a cifrelor.

Exemplu: Dacă $n=7$, iar șirul este **25 12 34 56 38 22 47**, atunci se afișează: Nr. cu suma cifrelor 7: 2, Nr. cu suma cifrelor 3: 1, Nr. cu suma cifrelor 11: 3, Nr. cu suma cifrelor 4: 1.

22 Rezultatele la teza de la matematică ale elevilor unei clase sunt înscrise într-o listă. Ordoneați lista descrescător și apoi afișați media pe clasă și numărul de elevi care au nota mai mare decât aceasta. Afișați și dacă există copii care au nota 5 la teză. Clasa are maxim 30 de elevi.

Exemplu: Dacă $n=5$, iar șirul este **10 8 9 10 5**, atunci se afișează: Rezultatele ordonate descrescător: **10 10 9 8 5**, Media pe clasă=8.4, Nr. elevi peste medie=3, Exista elevi cu 5.



23 Se dă un șir de n elemente numere naturale, unde $1 \leq n \leq 100$. Scrieți un program care:

- a ordonează crescător elementele pare ale acestui șir;
- b generează un alt șir care conține toate numerele prime din primul șir, îl afișează urmat de mesajul **Da**, dacă există în acest șir numărul 11, iar altfel urmat de **Nu**.

Exemplu: Dacă $n=6$, iar șirul este **4 11 34 12 31 17**, atunci se afișează: **a 4 12 34** **b 11 31 17** Da.

24 Se dă un vector a cu n elemente numere naturale mai mici decât 2000, unde $1 \leq n \leq 100$. Să se scrie un program care să construiască un alt vector b ale cărui elemente să fie în ordine: media aritmetică a elementelor șirului a , maximul șirului a , minimul șirului a , toate multiplicat de atâtea ori cât este ultima cifră a ultimului element al șirului a .

Exemplu: Dacă $n=3$, iar șirul este **4 5 6**, atunci se afișează **30 36 24**.

Răspunsuri (selectiv)

Unitatea I – Autoevaluare (p. 33)

1. A se vedea p. 12. 2. a, b, c, d, e, f. 3. $d - 1$; $a - 2$; $b - 3$; $c - 4$. 4. a, b, c, e, f. 5. a, c, d, e. 6. A se vedea p. 17. 7. a – DA; b – NU (butonul **Sortare** este în fila **Date**); c – DA. 8. a. comenzile și elementele, calcul, tabele, diagrame, texte (p. 15); b. calcul, Imprimare, imprimantă (p. 18); c. decizie, Excel, logice, două (p. 27). 9. A se vedea p. 18. 10. A se vedea p. 29.

Unitatea II – Autoevaluare (p. 55)

1. A se vedea p. 36. 2. a, b, c, d, e, f. 3. $c - 1$; $d - 2$; $a - 3$; $b - 4$. 4. a, b, d, e, f. 5. A se vedea p. 52. 6. A se vedea p. 46. 7. a – DA; b – DA; c – DA. 8. a. utilizator, motoarele, sugestiv, informații; b. subsolul, subsol, subsol, subsol, subsolul. 9. d. 10. A se vedea p. 43.

Unitatea III – Autoevaluare (p. 105)

1. b; 2. c; 3. 4.

4.

```
#include <iostream>
using namespace std;
int main()
{
    int a[100], n, i;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
        cin>>a[i];

    for (i=0; i<=(n-1)/2; i++)
        if (i!=n-1-i)
            cout<<a[n-1-i]<<' '<<a[i]<<' ';
        else cout<<a[i];
    return 0;
}
```

6.

```
#include <iostream>
using namespace std;
int main()
{
    int a[100], i, eEgal=1, n;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
    {
        cin>>a[i];
        if (a[i]%10!=3) eEgal=0;
    }
    if (eEgal==1) cout<<"Da";
    else cout<<"Nu";
    return 0;
}
```

5.

```
#include <iostream>
using namespace std;
int main()
{
    int v[12], i, M=0, l=0;

    for (i=0; i<=11; i++)
    {
        cin>>v[i];
        if (M<v[i]) {M=v[i]; l=i;}
    }
    switch (l)
    {
        case 0: cout<<"Luna ianuarie.";break;
        case 1: cout<<"Luna februarie.";break;
        case 2: cout<<"Luna martie.";break;
        case 3: cout<<"Luna aprilie.";break;
        case 4: cout<<"Luna mai.";break;
        case 5: cout<<"Luna iunie.";break;
        case 6: cout<<"Luna iulie.";break;
        case 7: cout<<"Luna august.";break;
        case 8: cout<<"Luna septembrie.";break;
        case 9: cout<<"Luna octombrie.";break;
        case 10: cout<<"Luna noiembrie.";break;
        case 11: cout<<"Luna decembrie.";
    }
    return 0;
}
```

7.

```

#include <iostream>
using namespace std;
int main()
{
    float a[100], m=1000000;
    int i, j, n, poz=0, aux;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
    {   cin>>a[i];
        if (a[i]<m) {m=a[i]; poz=i;}
    }
    for (i=0; i<=poz-2; i++)
        for(j=i+1; j<=poz-1; j++)
            if (a[i]<a[j])
            {   aux=a[i];
                a[i]=a[j];
                a[j]=aux;
            }
    for (i=poz+1; i<=n-2; i++)
        for(j=i+1; j<=n-1; j++)
            if (a[i]>a[j])
            {   aux=a[i];
                a[i]=a[j];
                a[j]=aux;
            }
    for (i=0; i<=n-1; i++)
        cout<<a[i]<<' ';
    return 0;
}

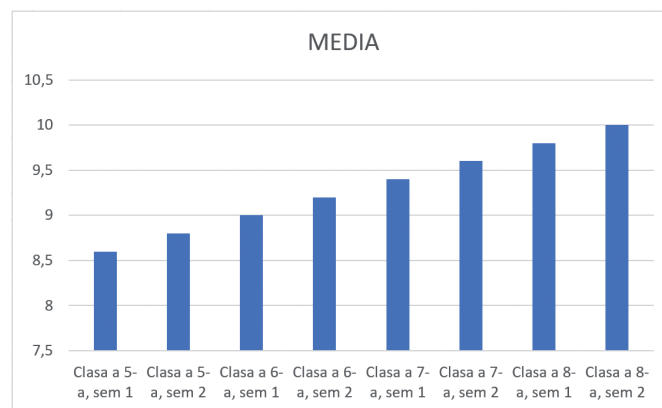
```

Recapitulare finală

1. b. 2. c. 3. a; c.

4.

	A	B
1	SEMESTRUL	MEDIA
2	Clasa a 5-a, sem 1	8,6
3	Clasa a 5-a, sem 2	8,8
4	Clasa a 6-a, sem 1	9
5	Clasa a 6-a, sem 2	9,2
6	Clasa a 7-a, sem 1	9,4
7	Clasa a 7-a, sem 2	9,6
8	Clasa a 8-a, sem 1	9,8
9	Clasa a 8-a, sem 2	10



5. c. 6. A se vedea p. 47. 7. Da. 8. a – F; b – F; c – A.
 9. c – 1; a – 2; b – 3. 10. a; b. 11. a[9]. 12. a. 13. b. 14. 0.
 15. b. 16. b. 17. c. 18. d. 19. a.

20.

```

#include <iostream>
using namespace std;

int main()
{
    int a[100], n, i, nr=0, S=0;
    int Mx=0, eGasit=0;
    float Ma=0;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
    {
        cin>>a[i];
        if (a[i]%33==0) nr++;
        if (a[i]%3==0) S=S+a[i];
        Ma=Ma+a[i];
        if (a[i]>Mx) Mx=a[i];
        if (a[i]==77) eGasit=1;
    }

    cout<<endl<<"a) ";
    for (i=0; i<=n-1; i=i+2)
        cout<<a[i]<<' ';

    cout<<endl<<"b) ";
    for (i=n-1; i>=0; i--)
        if (i%2==0) cout<<a[i]<<' ';

    cout<<endl<<"c) ";
    cout<<"Nr elem. divizibile cu 33="<<nr;

    cout<<endl<<"d) ";
    cout<<"Suma elem. multipli de 3="<<S;

    Ma=Ma/n;
    nr=0;
    for (i=0; i<=n-1; i++)
        if (a[i]>=Ma && a[i]<=Mx) nr++;

    cout<<endl<<"e) ";
    cout<<"Nr. elemente="<<nr;

    cout<<endl<<"f) ";
    if (eGasit==1) cout<<"Exista";
    else cout<<"Nu exista";
    return 0;
}

```

21.

```

#include <iostream>
using namespace std;

int main()
{
    int a[100], n, i, j, nr, S=0;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
    {
        cin>>a[i];

        //calculam suma cifrelor si o
        //introducem intr-un alt tablou
        nr=a[i];
        S=0;
        while (nr>0)
        {
            S=S+nr%10;
            nr=nr/10;
        }
        //inlocuim elementul cu valoarea
        //sumei cifrelor sale
        a[i]=S;
    }

    //calculam cate numere au aceeasi
    //suma a cifrelor, elementul deja
    //procesat il inlocuim cu -1
    for (i=0; i<=n-1; i++)
    {
        nr=1;
        if (a[i]!=-1)
        {
            for (j=i+1; j<=n-1; j++)
                if (a[j]!=-1 && a[i]==a[j])
                {
                    nr++;
                    a[j]=-1;
                }
            cout<<"Nr. cu suma cifrelor "<<
                a[i]<<": "<<nr<<endl;
            a[i]=-1;
        }
    }

    return 0;
}

```

22.

```

#include <iostream>
using namespace std;

int main()
{
    int a[30], n, i, j, m, nr=0, eGasit=0;
    float Ma=0;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
    {
        cin>>a[i];
        Ma=Ma+a[i];
        if (a[i]==5) eGasit=1;
    }

    //ordonare descrescatoare prin selectie
    for (i=0; i<=n-2; i++)
        for (j=i+1; j<=n-1; j++)
            if (a[i]<a[j])
            {
                m=a[i];
                a[i]=a[j];
                a[j]=m;
            }
    cout<<"Rezultatele ordonate: ";
    for(i=0; i<=n-1; i++)
        cout<<a[i]<<" ";

    Ma=Ma/n;
    cout<<endl<<"Media clasei="<<Ma<<endl;

    for(i=0; i<=n-1; i++)
        if (a[i]>Ma) nr++;
    cout<<"Nr elevi peste medie="<<nr<<endl;

    if (eGasit==1) cout<<"Exista elevi cu 5";
    else cout<<"Nu exista elevi cu 5";

    return 0;
}

```

23.

```

#include <iostream>
using namespace std;

int main()
{
    int a[100], b[100], n, i, j, k=0, m,
    nrDiv=0, eGasit=0;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
        cin>>a[i];

    cout<<"a) ";
    //ordonare crescatoare prin selectie
    for (i=0; i<=n-2; i++)
        if (a[i]%2==0)
        {
            for (j=i+1; j<=n-1; j++)
                if (a[j]%2==0)
                {
                    if(a[i]>a[j])
                    {
                        m=a[i];
                        a[i]=a[j];
                        a[j]=m;
                    }
                }
            cout<<a[i]<<' ';
        }

    for (i=0; i<=n-1; i++)
    {
        nrDiv=0;
        for(j=2; j<=a[i]/2; j++)
            if (a[i]%j==0) nrDiv++;
        if (nrDiv==0)
        {
            b[k]=a[i];
            if (b[k]==11) eGasit=1;
            k++;
        }
    }

    cout<<endl<<"b) ";
    for(i=0; i<=k-1; i++)
        cout<<b[i]<<" ";

    cout<<endl;
    if (eGasit==1) cout<<"Da";
    else cout<<"Nu";
    return 0;
}

```


24.

```
#include <iostream>

using namespace std;

int main()
{
    int a[100], b[3], n, i, c, M=0, m=2000;
    float Ma=0;

    cout<<"n="; cin>>n;
    for (i=0; i<=n-1; i++)
    {
        cin>>a[i];
        Ma=Ma+a[i];
        if (a[i]<m) m=a[i];
        if (a[i]>M) M=a[i];
    }

    Ma=Ma/n;
    c=a[n-1]%10;

    b[0]=Ma*c;
    b[1]=M*c;
    b[2]=m*c;

    for(i=0; i<=2; i++)
        cout<<b[i]<<" ";

    return 0;
}
```

pag. 7: Dreamstime;
pag. 8: Dreamstime.

Unitatea 1

pp. 10-11: Dreamstime;
pag. 13 – Știați că...: Wikimedia Commons;
pag. 21 – Amintiți-vă: Dreamstime;
pag. 24 – Știați că...: Dreamstime;
pag. 25 – Știați că...: Dreamstime;
pag. 26 – Descoperiți: Dreamstime;
pag. 29 – Descoperiți: Dreamstime;
pag. 30 – Știați că...: Wikipedia.

Unitatea 2

pp. 34-35: Dreamstime;
pag. 37 – Știați că...: Wikimedia Commons;
pag. 44 – Exersați: Dreamstime;
pag. 46 – Descoperiți: Dreamstime.

Unitatea 3

pp. 56-57: Dreamstime;
pag. 59 – Descoperiți: Wikipedia;
pag. 60 – Știați că...: Wikipedia;
pag. 61 – Exersați: Dreamstime;
pag. 63 – Știați că...: Wikimedia Commons;
pag. 64 – Descoperiți: Freepik, Pixabay;
pag. 65 – Exersați: Dreamstime;
pag. 67 – Aplicați și investigați: Dreamstime;
pag. 71 – Exersați: Dreamstime;
pag. 72 – Descoperiți: Dreamstime;
pag. 74 – Aplicați și investigați: Dreamstime;
pag. 75 – Exersați: Dreamstime;
pag. 78 – Descoperiți: Dreamstime; Știați că...: Wikimedia Commons;
pag. 79 – Exersați: Dreamstime;
pag. 80 – Descoperiți: Dreamstime;
pag. 81 – Descoperiți: Dreamstime;
pag. 82 – Descoperiți: Dreamstime;
pag. 83 – Exersați: Dreamstime;
pag. 85 – Descoperiți: Pixabay;
pag. 87 – Descoperiți: Dreamstime; Știați că...: Dreamstime;
pag. 88 – Descoperiți: Dreamstime;
pag. 89 – Descoperiți: Dreamstime;
pag. 93 – Descoperiți: Dreamstime;
pag. 94 – Descoperiți: Dreamstime;
pag. 95 – Descoperiți: Dreamstime;
pag. 98 – Descoperiți: Dreamstime;
pag. 99 – Exersați: Dreamstime;
pag. 101 – Exersați: Dreamstime;
pag. 100 – Descoperiți: Wikipedia;
pag. 103 – Autoevaluare: Dreamstime;
pag. 104 – Recapitulare finală: Dreamstime;
pag. 106 – Recapitulare finală: Dreamstime.

Programa școlară poate fi accesată la adresa <http://programe.ise.ro>

Informatică și TIC

```
int n="], m, v[100  
i=0; ; cin>>n;  
cin> ; i++)  
(i=0;  
for (]=i+1; j  
if(v[i]>v[j])  
{  
    v[i]=v[j];  
    ]=m;  
}  
i=0; i<=n-1.
```

Clasa a VIII-a

$m = v[i]$