

INSPECTORATUL ȘCOLAR JUDEȚEAN GALAȚI

---

ASOCIAȚIA PROFESORILOR DE  
INFORMATICĂ "TUDOR SORIN" DIN GALAȚI

**REVISTA DE INFORMATICĂ**

**(iM)PULS ++**



Numărul 1, online ISSN-L 2067 - 9815  
Numărul 12, ISSN 2601 - 5951, Data: 12 aprilie 2018



## **„Informatică și TIC pentru gimnaziu**

**N**oua Programă școlară pentru disciplina INFORMATICĂ ȘI TIC, clasele a V-a – a VIII-a, încurajează elevii să folosească tehnologia într-un mod responsabil și creativ și răspunde nevoilor de continuare a eforturilor de alfabetizare digitală și de reconsiderare a acestui concept din perspectiva noilor cerințe socio-profesionale. Acest lucru se face printr-un curriculum relevant, vizând formarea unui set de competențe digitale, pe care fiecare absolvent să le poată valorifica pe parcursul școlarității și în viața activă. Programa de Informatică și TIC pentru clasele V-VIII a fost realizată pe baza unor documente relevante de cercetare și politici la nivel internațional, european și național care au contribuit la definirea competențelor generale și specifice necesare a fi dezvoltate până la finalul clasei a VIII-a și a abordării metodologice specifice. Programa de Informatică și TIC pentru clasele V-VIII răspunde nevoilor de continuare a eforturilor de alfabetizare digitală și de reconsiderare a acestui concept din perspectiva noilor cerințe socio-profesionale, printr-un curriculum relevant, vizând formarea unui set de competențe digitale, pe care fiecare absolvent să le poată valorifica pe parcursul școlarității și în viața activă. Conform planului-cadru pentru învățământul gimnazial, aprobat prin OMENCS nr. 3590/2016, disciplina Informatică și TIC se predă în clasele a V-a, a VI-a, a VII-a și a VIII-a, ca disciplină de trunchi comun, cu o alocare de o oră/săptămână.

### **Platforme realizate de profesori pentru profesori**

<http://www.informaticainscoli.ro>

<http://www.infogim.ro/>

Comunitatea InfoGim se adresează tuturor celor interesați de informatica și tehnologia informațiilor predate în ciclul gimnazial. Aici sunt resurse necesare predării și învățării în sprijinul elevilor profesorilor și părinților ce vor beneficia de această nouă disciplină modernă și practică.

<https://infotiegim.wordpress.com/>

<http://www.predauviitor.ro/despreprogram/>

Fiecare elev din România să beneficieze de o educație informatică bazată pe proiecte și descoperire, trans-disciplinară și aplicată, facilitată de profesori de informatică conectați cu realitatea și viitorul tehnologiei și pregătiți pedagogic să faciliteze elevilor experiențe de învățare autonomă.

<https://www.consumerclassroom.eu/ro/partenerii-proiectului>

Site-ul Consumer Classroom are ca scop promovarea educației consumatorului și, în special, încurajarea predării acestei discipline în cadrul învățământului secundar din Europa. Este un site pan-european multilingvistic, finanțat de Comisia Europeană.

Punctul forte al site-ului constă în calitatea resurselor didactice și în instrumentele de colaborare ca de exemplu aplicația pentru crearea lecțiilor, forumuri și chat-ul live. Este indicat profesorilor care predau elevilor cu vârste între 12 și 18 ani fiind, totodată, deschis tuturor celor care doresc să contribuie. Site-ul a fost dezvoltat pentru a facilita profesorilor posibilitățile de a afla despre subiecte legate de consumatori, de a crea sau opta pentru lecții gata concepute pe care să le predea elevilor și de a colabora online cu alți profesori și clase din Europa.

Profitând de cele mai noi tehnologii web și de utilizarea internetului de către comunități, Consumer Classroom furnizează resurse didactice gata de utilizare cu privire la o gamă largă de teme legate de educația consumatorului, de la consumul durabil și responsabil până la inițierea în domeniul financiar.

Consumer Classroom invită profesorii să devină parte a unei comunități web dinamice de profesori din întreaga Europă și nu numai. Îi ajută să împărtășească informațiile cu elevii, încurajează proiectele multidisciplinare interșcolare și, de asemenea, oferă profesorilor oportunitatea de a deveni experți în educația consumatorului pe site-ul web.

Ambiția Consumer Classroom este de a deveni o voce competentă în educația online a consumatorului pentru a contribui la îmbunătățirea gradului de conștientizare a adolescenților, legitimându-ne viitorii consumatori atât la nivel național cât și european.

**prof. Musunoiu-Novetschi Iuliana-Monica**  
**informatica si managementul resurselor umane**  
**Inspector școlar, ISJ Galați**



## APLICAȚIA LinkLabel

Deschideți mediul integrat de programare:

**Start→All Programs→Visual C# 2008 Express Edition**

În meniul **File**, selectați **New Project**. Se deschide fereastra de dialog **New Project** din care se selectează **Windows Forms Application**. În câmpul Name se scrie numele aplicației (de exemplu Interfata), click **OK**.

Se crează un proiect de tip **Windows Form**.

Din **Toolbox** se aleg următoarele **controale**:

2 controale de tip Button

Un control de tip Button - se scrie la **Text**: "Deschide fereastra" – button 1

Un control de tip Button - se scrie la **Text**: "Închide" – button 2

Se tratează evenimentul Click corespunzător butonului 1 și se scrie codul:

```
Form2 form2=new Form2();  
this.Hide();  
form2.ShowDialog();
```

### Observații

*Fereastra Form2 este activată cu metoda ShowDialog, metodă care permite ca revenirea în fereastra din care a fost activat noul formular să se facă doar după ce noul formular a fost închis (noul formular, form2, a fost deschis modal). Pentru a ascunde formularul a fost setată proprietatea Visible indirect: **this.Hide()** sau se putea scrie **this.Visible=false**;*

Se tratează evenimentul Click corespunzător butonului 2 și se scrie codul:

```
Application.Exit(); //Ieșire din aplicație
```

Din meniul **Project** selectați **Add Windows Form**, iar în caseta de dialog care apare adăugați numele **Form2**, pentru noul formular creat. În acest moment ați inclus în program două formulare. Din **Toolbox**, trageți cu mouse-ul pe suprafața formularului:

Un control de tip Button - se scrie la **Text**: "Închide"

Un control de tip **LinkLabel** – se scrie la **Text**: "Deschide fișier!"

Se tratează evenimentul Click corespunzător butonului button1 și se scrie codul:

```
Form1 form1=new Form1();  
this.Close();  
form1.Visible = true;
```



**Temă: Descrie pe caiet codul corespunzător evenimentului Click al butonului1 din form2.**

Se tratează evenimentul LinkClicked corespunzător controlului LinkLabel și se scrie codul:

```
try
{
    // se crează instanța unui proces
    System.Diagnostics.Process myprocess = new System.Diagnostics.Process();
    // se scrie calea de director a executabilului care va fi lansat în
    execuție
    mypro-
    cess.StartInfo.FileName=@"C:\Users\MadalinaP\Desktop\test\bin\Debug\test.exe";
    // se cere permisiunea de a rula procesul
    myprocess.StartInfo.Verb = "runas";
    // se folosește metoda Start() pentru a deschide fisierul
    myprocess.Start();
}
catch (Exception ex)
{
    //în cazul în care fișierul nu a fost găsit
    MessageBox.Show("Fisierul nu exista");
}
```

### Observații

Blocul try – catch() este o excepție și în esență este un **eveniment** care se produce în timpul execuției unui program și care poate **perturba** fluxul normal al instrucțiunilor acestuia. De exemplu, în cadrul unui program în care se dorește deschiderea unui fișier, astfel de evenimente excepționale pot fi:

*Absența fișierului pe care dorim să-l deschidem*

*Imposibilitatea de a-l deschide din diferite cauze (permisiuni insuficiente), etc*

Se observă includerea instrucțiunilor ce aparțin fluxului normal de execuție într-un bloc try, și precizarea condițiilor excepționale posibile la sfârșit, în câte un bloc catch. **Logica** este următoarea: se execută instrucțiune cu instrucțiune secvența din blocul try și, la apariția unei situații excepționale, semnalate de o instrucțiune, **se abandonează** restul instrucțiunilor rămase neexecutate și se **sa-re** direct la blocul catch corespunzător.

**Exception** este o clasă care desemnează comportamentul specific unei excepții.



**Temă: Documentează-te și notează despre clasa **Exception****

Compilați și rulați cu **F5**



## Magia numerelor Fibonacci

**Prof. Ioniță Doina**

**Colegiul Național "Calistrat Hogaș" Tecuci**

Șirul lui Fibonacci:

Start: 0 și 1

$0 + 1 = 1$   
 $1 + 1 = 2$   
 $2 + 1 = 3$   
 $3 + 2 = 5$   
 $5 + 3 = 8$   
 $8 + 5 = 13$   
 $13 + 8 = 21$   
 $21 + 13 = 34$   
 $34 + 21 = 55$   
 $55 + 34 = 89$   
 $89 + 55 = 144$   
 $144 + 89 = 233$   
 $233 + 144 = 377$   
 $377 + 233 = 610$   
 $610 + 377 = 987$   
 $987 + 610 = 1597$   
 $1597 + 987 = 2584$   
 $2584 + 1597 = 4181$   
 $4181 + 2584 = 6765$   
 $6765 + 4181 = 10946$   
  
 0, 1, 1, 2, 3, 5, 8, 13,  
 21, 34, 55, 89, 144,  
 233, 377, 610, 987,  
 1597, 2584, 4181,  
 6765, 10946...

Raportul de aur, sau  
numărul  $\Phi = 1,61803...$

$2 : 1 = 2$   
 $3 : 2 = 1,5$   
 $5 : 3 = 1,6666$   
 $8 : 5 = 1,6$   
 $13 : 8 = 1,625$   
 $21 : 13 = 1,6153$   
 $34 : 21 = 1,6190$   
 $55 : 34 = 1,6176$   
 $89 : 55 = 1,6181$   
 $144 : 89 = 1,6179$   
 $233 : 144 = 1,61803$   
 $377 : 233 = 1,61803$   
 $610 : 377 = 1,61803$   
 $987 : 610 = 1,61803$   
 $1597 : 987 = 1,61803$   
 $2584 : 1587 = 1,61803$   
 $4181 : 2584 = 1,61803$   
 $6765 : 4181 = 1,61803$   
 $10946 : 6765 = 1,61803$

Numărul  $\Phi$  se obține  
împărțind un element al  
Șirului lui Fibonacci la  
precedentul său.

Cine nu a observat și nu a rămas plăcut surprins câtă simetrie și ordine există în natură? Poate că mulți dintre noi deja am dedus că natura a folosit în „hazardul” ei formule matematice ce au creat în final tot ceea ce ne înconjoară. Oamenii încearcă permanent să înțeleagă natura și legile acesteia, să simtă ritmurile cosmice, să înțeleagă de fapt mai profund viața, pentru a ajunge la o armonie cu mediul înconjurător.

Din antichitate, numerele au stat la baza simbolisticii, exprimând forțe și idei. Platon le considera cea mai înaltă treaptă a cunoașterii și esența armoniei cosmice și interioare. Acesta afirma că „totul este potrivit după număr”. Pitagora susținea că întreg universul este ordonat matematic și orice poate fi exprimat prin numere.

Privind în jur, ne putem imagina că ramurile, frunzele și florile cresc la întâmplare. Adevărul este totuși că locurile în care apare fiecare ramură, frunză, tulpină, mugur sau petală nu este întâmplător, ele dezvoltându-se în conformitate cu niște „legi” precise ale naturii. Există anumite tipare ori-

unde ne-am uita în natură, cel mai des întâlnit fiind însă Șirul lui Fibonacci.

Dar cine este Fibonacci?

Leonardo Pisano Fibonacci sau Leonardo din Pisa (1170-1250), este considerat ca unul dintre cei mai mari matematicieni europeni ai Evului Mediu. S-a născut în Pisa, oraș italian faimos pentru turnul său înclinat. Cartea lui Fibonacci „Liber Abaci”, publicată la începutul secolului al XIII-lea, a introdus această secvență magică în lumea occidentală.

Secvența Fibonacci este atât de simplă.... Aici, fiecare număr este creat prin însumarea ultimelor două numere, începând de la 1 1 2 3 5 8 13 21 ... și până la infinit.

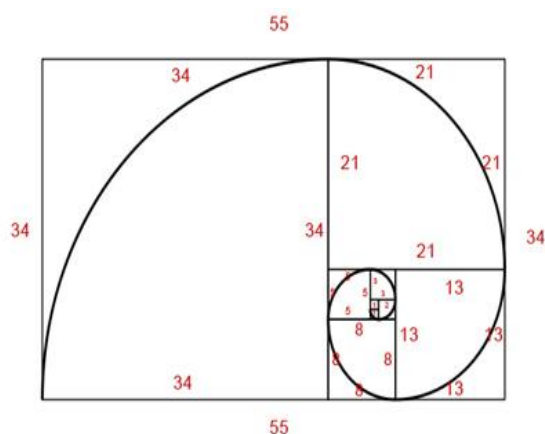
Șirul astfel obținut, în care fiecare termen este suma celor două numere precedente, s-a dovedit a fi extrem de util în rezolvarea anumitor probleme și el apare în multe situații în matematică și știință în general.

Secvența Fibonacci este atât de des întâlnită în natură încât este o provocare să găsești o plantă sau o structură de fructe care nu este conformă cu aceasta. De exemplu, modul în care sunt așezate frunzele de-a lungul unei tulpini este guvernată de secvența Fibonacci, asigurându-se că fiecare frunză are acces maxim la lumina soarelui și la ploaie. Același principiu este și în cazul formării conurilor de pin, a florii-soarelui, a ananasului sau a cactușilor. Toate plantele sunt geometrice într-un fel sau altul. Cu toate acestea, există plante a căror geometrie este mai pronunțată decât altele. Șirul lui Fibonacci reprezintă totuși mai mult decât o sursă de amuzament. Dacă împărțim oricare dintre numerele lui Fibonacci la următorul număr mai mare din serie, după cifra 3, răspunsul este întotdeauna 0,61. După 89, răspunsul este întotdeauna 0,618; după numerele mai mari, se pot completa mai multe zecimale. Împărțiți oricare număr la numărul care îl precedă. După 2, răspunsul este întotdeauna 1,6. După 144, răspunsul este întotdeauna 1,618.





Grecii cunoșteau acest raport și îl nu-



Șirul lui Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946...

meu „proporția divină” sau „numărul de aur”. Acest număr definește proporțiile Parthenonului, forma cărților de joc și a cardurilor bancare, precum și proporțiile clădirii Adunării Generale a ONU din New York.

Să facem o incursiune în istoria numerelor și a proporțiilor, în lumea artei și în natură, având ca punct comun Proporția de Aur. Elementul comun ce unește știința, matematica, arta, natura, domenii care aparent nu pot fi relaționate este ”Numărul/Proporția de Aur”.

Dacă privim mâinile unui om, constatăm alte coincidențe, ce ne amintesc de faimosul

șir. Avem 2 mâini, cu câte 5 degete, fiecare având 3 falange separate prin două articulații.

Coincidență sau nu, aspectul este interesant, cu atât mai mult cu cât dacă măsurăm lungimea oaselor degetelor, se pare că raportul dintre osul cel mai lung și cel din mijloc, ca și raportul dintre osul mijlociu și cel mai scurt din vârf reprezintă Proporția de Aur Phi.

În medie, dimensiunile falangelor sunt: 2 cm, 3 cm, 5 cm, iar în continuare osul palmei are circa 8 cm (2, 3, 5, 8 sunt numere din secvența Fibonacci). În același timp, fața umană este caracterizată din punct de vedere estetic prin câteva dimensiuni principale: distanța dintre ochi, distanța dintre gură și ochi, distanța dintre nas și ochi, dimensiunea gurii.

Proporția de Aur este peste tot, dar nicăieri nu este la fel

de impresionantă și la fel de importantă decât ceea ce descoperim în corpul uman, mai important ca ADN-ul nostru. Fiecare ciclu al moleculei de ADN măsoară 34 angstromi lungime și 21 de angstromi lățime.

Precum știm deja, 34 și 21 sunt numere Fibonacci, așa cum s-a menționat mai sus, 34/21 este în raport cu  $\Phi = 1.619$ .

Aplicațiile numărului de aur, de fapt ale raportului, se regăsesc la punerea în proporție

a lucrărilor în arhitectură, pictură, sculptură, estetică și artă în general, ceea ce confirmă interesul manifestat de-a lungul timpului pentru acest număr. Spre exemplu se consideră că fața Giocondei lui da Vinci se încadrează într-un astfel de dreptunghi.

Proporția divină a condus la construirea Dreptunghiului de aur, în care raportul laturilor este egal cu numărul de aur. Acest tip de dreptunghi este considerat ca fiind deosebit de estetic și ca urmare a fost și este intens utilizat în arhitectură și artă.

Litera grecească  $\phi$  phi simbolizează Secțiunea Divină, acest  $\phi$  este mai ușor de vizualizat în formă geometrică și e reprezentat de un dreptunghi format din alipirea unui pătrat și a unui dreptunghi a cărui proporție este de 1 la 1,618 față de pătratul respectiv. Această operațiune poate fi repetată în ambele sensuri la infinit. Dacă trasăm un sfert de cerc în fiecare pătrat pe măsură ce latura acestuia crește, obținem Spirala de Aur (sau Spirala Divină).

Mai puțin cunoscut este faptul că Spirala de Aur este cel mai simplu tipar matematic natural. În acest tipar se înscriu cochiliile melcilor, scoicile, coarnele animalelor, florile, plantele, etc. Proporția Divină nu este singurul model matematic care tinde spre infinit, mai există multe asemenea modele. Știind aceasta, ne punem întrebarea: cum ar putea infinitul să se repete? Dacă ceva se întâmplă la infinit, cum ar putea să se întâmple de 2 ori? Infinitul reprezintă ceea ce e etern, ceea ce e cu adevărat întreg.

Unicitatea fiecărui lucru este doar ceea ce știm, ce percepem și de ce suntem conștienți, este universul privindu-se pe sine. Acesta este universul și noi suntem parte din el.

**„Lucrurile sunt grele până li se descoperă taina, pe urmă totu-i ușor.”  
- Pedro Calderon de la Barca.**

## PROIECTE LA GEOGRAFIE

**Enunț:** Profesoara de geografie dorește să distribuie teme de referat elevilor săi din clasa a XI-E. Dumneaei a pregătit p teme celor n elevi,  $n \geq p$ , și trebuie ca fiecare temă să fie luată cel puțin o dată, indiferent dacă doi sau mai mulți elevi își vor lua aceeași temă.

Valorile n și p se vor citi de la tastatură, iar rezultatele trebuie scrise în fișierul “geografie.out”, sub forma dată în exemplu, toate modurile de distribuire a proiectelor, precum și numărul de soluții.

Tastatura	geografie.out
n = 3	1 1 2
k = 2	1 2 1
	1 2 2
	2 1 1
	2 1 2
	2 2 1
	sunt 6 soluții

### Analiza problemei și rezolvarea ei.

Spunem că o funcție  $f: A \rightarrow B$  este surjectivă dacă oricare ar fi  $y \in B$ , există  $x \in A$  astfel încât  $f(x) = y$ , adică  $f(A) = B$ . Rezolvarea problemei constă în generarea tuturor funcțiilor surjective definite pe mulțimea  $\{1, 2, \dots, n\}$  cu valori în mulțimea  $\{1, 2, 3, \dots, k\}$ . Pentru generarea acestor funcții vom folosi algoritmul backtracking.

Pentru validarea unei combinații drept soluție parțială, vom utiliza un vector **viz** care memorează de câte ori elementul i din mulțimea B a fost pus în combinație. Vom mai utiliza și un contor nr care determină câte elemente distincte din A au fost utilizate până la pasul k. Pentru ca o combinație să fie soluție parțială, trebuie ca numărul elementelor diferite adăugate în combinația parțială plus numărul elementelor rămase de pus să fie  $\geq$  cu numărul elementelor din B.

Observație: Dacă A are n elemente și B are p elemente, atunci numărul funcțiilor surjective definite pe A cu valori în B este:

$$n^k - C_n^1(n-1)^k + C_n^2(n-2)^k + \dots + (-1)^{n-1} C_n^{n-1}$$



```

//functii surjective
#include<iostream>
#include<fstream.h>
using namespace std;
fstream f("geografie.out",ios::out);
int st[10],n,k,as,ev,p;

void init()
{st[k]=0;
}

int sucesor()
{if (st[k]<p)
    {st[k]=st[k]+1;
    return 1;
    }
else
    return 0;
}

int valid()
{//in viz se memoreaza de cate ori ele-
mentul i din B a fost vizitat
//in nr determinam cate elemente distinc-
te au fost puse in stiva
int viz[10]={0};
int nr=0,i;
for (i=1;i<=k;i++)
    viz[st[i]]++;
for(i=1;i<=k;i++)
    if (viz[i]!=0)
        nr++;
/*pt ca o combinatie sa fie solutie
partiala, trebuie ca nr elementelor diferi-
te
adaugate in combinatia partiala plus
numarul elementelor de pus sa fie >=
cu numarul elementelor din B*/
return nr+n-p>=k;
}

```

```

int solutie()
{return k==n;
}

void tipar()
{int i;
for (i=1;i<=n;i++)
    f<<st[i]<<' ';
f<<endl;}

int main()
{int nrsol=0;
cout<<"numarul de elevi:"; cin>>n;
cout<<"numarul de teste:"; cin>>p;
k=1; init();
while (k>0)
    {as=1; ev=0;
    while (as && !ev)
        {as=sucesor();
        if (as)
            ev=valid();
        }
    if (as)
        if (solutie())
            {
                tipar();
                nrsol++;
            }
        else
            {k++;
            init();
            }
        else
            k=k-1;
    }
f<<"sunt "<<nrsol<<" solutii";
f.close();
return 0;
}

```





## Cum funcționează Instagram? Filtre și operatori de imagine

Irimia Robert și Rusu Miruna, elevi clasa a XII-a, CNVA

O idee foarte vehiculată în ziua de astăzi este aceea că tehnologia ocupă cel mai important loc în viața noastră, ceea ce este și destul de adevărat. Trăim în generația gadgeturilor, a aplicațiilor și a programelor, în care nu mai putem deosebi clar ce este "raw" și ce este trecut prin filtrul tehnologiei. O amprentă foarte profundă pe care aceasta o are asupra vieții noastre se regăsește în comunicare. Socializarea a devenit mult mai accesibilă, indiferent de timp sau de spațiu, în ultimele decenii: indiferent cât de mare este distanța dintre tine și un prieten de-al tău, sau un profesor, sau chiar o pisică cu un cont de Instagram sau Facebook, un simplu mesaj creează o conexiune. Dar mesajul aparține zilei de ieri... Cu cât avansăm mai mult, cu atât ne dorim o comunicare cât mai complexă, o experiență aproape "reală". Ei bine, pentru această dorință de apropiere, de real, a apărut Instagram, aplicație folosită de milioane de utilizatori în ziua de azi, care surprinde în fiecare zi cu un nou Update: prima oară video-urile, apoi live-urile, story-urile, iar mai nou, filtrele de imagine, foarte numeroase și diversificate de altfel, pe care le folosim mereu când vrem să scăpăm de simplul și obișnuitul imaginilor. Numai că această dorință de evadare din "simplu" ascunde, de fapt, un proces destul de complicat. Despre asta urmează să vă povestesc: filtrele de imagine și ce se ascunde în spatele lor.

Pentru a înțelege cum manipulează un program imaginile, trebuie să înțelegem mai întâi ce reprezintă acestea pentru calculator. Vom discuta un format de imagine foarte comun, PNG, care admite și valoare de transparență. Astfel, imaginile PNG au o adâncime a culorii de 32 de biți, adică fiecare culoare este reprezentată de 32 de biți în memorie. În total, PNG poate reține  $2^{32}$  culori.

Având în vedere că citiți o revistă de informatică, voi presupune că sunteți familiari cu noțiunea de matrice (tablou bidimensional). În acest caz, înțelegerea reprezentării unei poze în memoria calculatorului devine extrem de facilă; pozele sunt pur și simplu niște matrici de pixeli, un pixel fiind un pătrățel de o anumită culoare. Dacă dați zoom foarte mult pe o poză, veți putea observa pixelii dispuși pe rânduri și coloane, exact ca elementele unei matrici.

Având în vedere că ochiul uman are receptori pentru culorile roșu, verde și albastru, culorile sunt reținute în memorie astfel: 8 biți pentru canalul roșu, 8 pentru verde, 8 pentru albastru, iar restul de 8 biți sunt rezervați pentru transparență (numit canal alpha). Așa arată culoarea stocată de un pixel în memorie:

Sample Length:	8								8								8								8							
Channel Membership:	Alpha								Red								Green								Blue							
Bit Number:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Wikipedia- RGBA Color Space

Modelul prezentat mai sus se numește ARGB, după ordinea în care se stochează cele 4 componente. Mai puțin folosit, deși există, este modelul RGBA, însă ne vom concentra pe primul, deoarece principiile de folosire a celor 2 standarde se păstrează.

Acum că ați înțeles structura datelor cu care lucrăm, hai să vedem cum sunt ele folosite mai exact pentru a aplica filtre și operatori. Trebuie reținut, însă, că având în vedere dimensiunea de 32 de biți a unui pixel, aceștia se memorează în majoritatea limbajelor de programare în variabile de tip "int", deci ca numere întregi de 32 de biți. Deci, pentru început, să vedem cum putem extrage fiecare canal de culoare într-o variabilă separată pentru a putea fi prelucrat. Pentru aceasta, cel mai des se folosește operatorul pe biți "AND" binar, care în C/C++, Java, C# și alte limbaje bazate pe această sintaxă, este reprezentat de semnul "&". Atenție, a nu se confunda cu operatorul logic "&&";!!!



Cum nu este un operator folosit des în liceu, haideți să ne amintim rapid ce face acesta. Sintaxa pentru folosirea lui este "valoare\_1 & valoare\_2", iar operatorul va aplica operația "AND" pe fiecare bit din valoare\_1 și pe bitul corespunzător din valoare\_2. De exemplu, rezultatul operației 0b01101010 & 0b01011010 va fi 0b01001010. Prefixul "0b" este folosit pentru a arăta că urmează un număr reprezentat în baza 2. De asemenea se mai poate folosi "0x", pentru baza 16.

Pentru a extrage valoarea componentei albastre (B-ul), vom folosi expresia următoare:

Blue = pixel & 0b11111111

Dacă nu este clar de ce, gândiți-vă că în fața celor 8 de 1 vin 24 de 0-uri astfel:

Blue = pixel & 0b00000000000000000000000011111111

Astfel, ultimii 8 biți din pixel vor fi păstrați neschimbați, iar tot restul vor deveni 0, pentru că îi operăm cu 0-uri.

Acum, înlocuind notația binară cu notație hexadecimală pentru a fi mai ușor de citit, vom avea:

Blue = pixel & 0xFF

Green = pixel & 0xFF00

Red = pixel & 0xFF0000

Alpha = pixel & 0xFF000000

Acum că știm cum să manipulăm pixelii, hai să trecem la primul filtru despre care vom discuta astăzi, și anume cel Grayscale. Mai exact, vom folosi 3 tipuri de filtrare Grayscale. Înainte de asta, dacă nu ați fost atenți la orele de desen, vă voi aminti că nuanța de gri apare atunci când toate culorile pe care le recepționează ochiul sunt în proporții egale (Red = Green = Blue). Astfel, valoarea de gri calculată va fi atribuită tuturor celor 3 componente ale spațiului de culoare.

#### Grayscale median

Posibil cel mai comun algoritm, valoarea de gri a unui pixel este calculată prin formula:

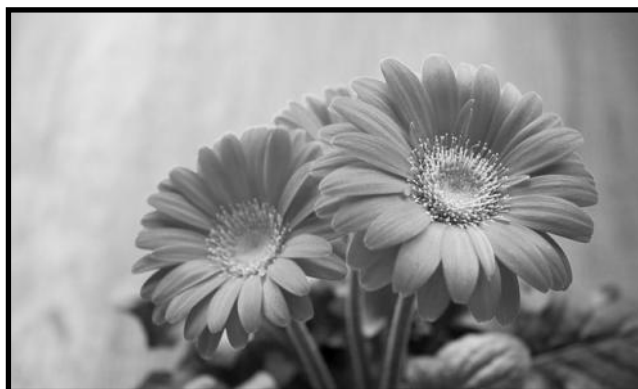
Grey = (Red + Green + Blue) / 3

Simplu și eficient, însă nu dă rezultate foarte plăcute din punct de vedere estetic.

#### Grayscale Luma

Acesta este considerat a genera cele mai frumoase efecte de grayscale, deoarece ia în considerare faptul că ochiul uman nu percepe cele 3 componente ale culorii în mod egal (receptorii de la nivelul ochiului sunt mai sensibili la roșu decât la albastru și verde, de exemplu):

Grey = 0.59 \* Red + 0.3 \* Green + 0.11 \* Blue



#### Grayscale prin desaturarea culorii

Posibil cea mai corectă din punct de vedere matematic, aceasta transformă culoarea din spațiul RGB în spațiul HSV, acronim pentru "Hue, Saturation, Value". Hue reprezintă nuanța culorii (galben, violet, etc) și din punct de vedere geometric, arată ca o roată cu toate culorile dispuse radial în jurul ei, roșu la 0 grade, verde la 120, albastru la 240. Saturația arată cât de "vie" este culoarea: o culoare foarte puternică are saturație mare, în timp ce griul are saturație 0. Value arată cât de închisă sau deschisă este o culoare, albul având luminozitate maximă, iar negrul 0.



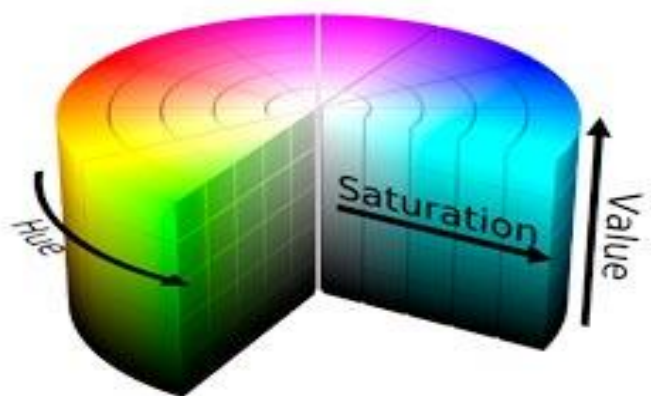


Figure 2

<http://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/colorreduction/>

HSL, HSB AND HSV codeitdown.com



Pentru că formula de conversie este relativ complexă, majoritatea limbajelor de programare au funcții care pot face conversia fără a trebui să o înțelegem. Astfel, după ce transformăm valoarea pixelului din RGB în HSV, pur și simplu îi forțăm saturația la 0, pentru a obține nuanța cea mai corectă de gri din punct de vedere matematic pentru culoarea respectivă. Apoi, transformăm înapoi în spațiul RGB.

Mai departe vom introduce noțiunea de "kernel". În procesarea de imagine, un kernel este o matrice pătratică de forma  $n \times n$ , cu  $n$  număr natural impar. Pentru a aplica un filtru/efect/operator unei imagini, se aplică o funcție fiecărui pixel în parte sau unor grupări de pixeli din poză, funcție care depinde de kernel-ul ales și care ia valori într-un spațiu de culoare (de obicei RGB).

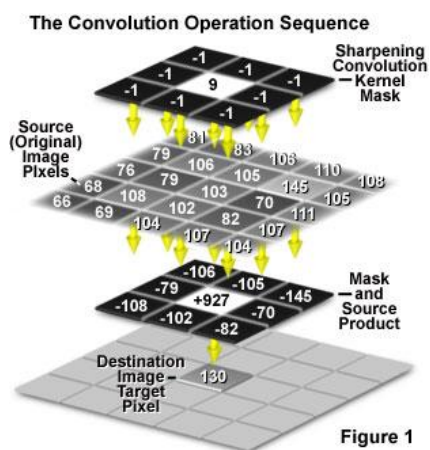


Figure 1

8	5	8	0	6	8	7
9	9	1	8	1	6	7
1	$i-1$	$i$	$i+1$	9	7	3
9	$i-1$	1	$i$	$i+1$	6	0
6	$i-1$	$i$	$i+1$	7	$i+1$	2
0	9	9	7	0	4	6
2	4	6	3	0	3	7

Pentru a explica folosirea acestei matrici, vom folosi imaginea de mai sus. Kernel-ul dat drept exemplu are dimensiunea de  $3 \times 3$ . Sub acesta sunt reprezentate valorile pixelilor dintr-o poză. De reținut că în general, un kernel

se aplică ori pentru fiecare canal în parte (R, G și B), pentru a obține imagini colorate, ori valorii de gri obținute prin conversia Grayscale, caz în care se obține o imagine alb-negru. Kernel-ul se aplică fiecărui pixel din imagine, suprapunând valoarea din mijlocul kernel-ului peste pixelul pe care vrem să îl procesăm, ca în imaginea 2. Apoi, înmulțim fiecare valoare din kernel cu valoarea corespunzătoare din imaginea procesată, facem suma acestor valori, iar rezultatul îl memorăm în pixelul corespunzător dintr-o imagine nouă (deoarece dacă am lucra direct pe imaginea ce trebuie procesată, pixelii ar fi alterați).





Cel mai comun exemplu de efect aplicat prin kernel este "Mean Blur". Kernel-ul poate avea orice dimensiune, iar toate valorile din acesta sunt  $1/(n * n)$ . Cu cât kernel-ul este mai mare, cu atât efectul de blur este mai puternic, pentru că face media mai multor pixeli din jur.

Exemplu de folosire a unui filtru de blur pentru a "repara" o imagine cu pixeli alterați.



Blur 3x3



Blur 7x7



Un alt efect foarte comun folosit este cel de sharpening, care mărește contrastul dintre pixelii din imagine. Kernel-ul folosit cel mai des este următorul:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

În principiu, majoritatea filtrelor folosite de Instagram sunt obținute ori prin aplicarea unui singur kernel asemănător celor de mai sus, ori prin aplicarea unei succesiuni de kernel-uri și operatori. Vă invităm să citiți mai multe despre acest subiect și să implementați chiar voi aceste metode de procesare a imaginii!



Pentru a vă ajuta, mai jos aveți codul cu ajutorul căruia am generat pozele din articol, scris în Python. Acesta folosește biblioteca Python Image Library (PIL) pentru a ne ușura munca. Puteți încerca, de asemenea, implementări în C++ pentru un timp de execuție mai mic.

```
import os, sys
import Image
nr_imagini = 2
for i in range(0, nr_imagini):
    im = Image.open("image" + (str)(i + 1) + ".png")
    pixels = im.load()

    gray1 = Image.new(im.mode, im.size)
    gray2 = Image.new(im.mode, im.size)
    gray3 = Image.new(im.mode, im.size)
    sharpen = Image.new(im.mode, im.size)
    blur3x3 = Image.new(im.mode, im.size)
    blur7x7 = Image.new(im.mode, im.size)

    pixels_gray1 = gray1.load()
    pixels_gray2 = gray2.load()
    pixels_gray3 = gray3.load()
    pixels_sharpen = sharpen.load()
    pixels_blur3x3 = blur3x3.load()
    pixels_blur7x7 = blur7x7.load()
    for k in range(0, im.size[0]):
        for j in range(0, im.size[1]):
            gray = (pixels[k, j][0] + pixels[k, j][1] + pixels[k, j][2]) / 3
            pixels_gray1[k, j] = (gray, gray, gray, 0)
    for k in range(0, im.size[0]):
        for j in range(0, im.size[1]):
            gray = int(pixels[k, j][0] * 0.59 + pixels[k, j][1] * 0.3 + pixels[k, j][2] * 0.11)
            pixels_gray2[k, j] = (gray, gray, gray, 0)
    for k in range(0, im.size[0]):
        for j in range(0, im.size[1]):
            gray = (min(pixels[k, j][0], pixels[k, j][1], pixels[k, j][2]) + max(pixels[k, j][0], pixels[k, j][1], pixels[k, j][2])) / 2
            pixels_gray3[k, j] = (gray, gray, gray, 0)
    blur7x7.save("Blur 7x7_" + str(i + 1) + ".png")
```

```

for k in range(1, im.size[0] - 1):
    for j in range(1, im.size[1] - 1):
        red = pixels[k, j][0] * 9 - pixels[k - 1, j - 1][0] - pixels[k, j - 1][0] - pixels[k - 1, j][0] - pixels[k + 1, j][0] - pixels[k + 1, j + 1][0] - pixels[k, j + 1][0] - pixels[k + 1, j - 1][0] - pixels[k - 1, j + 1][0]
        green = pixels[k, j][1] * 9 - pixels[k - 1, j - 1][1] - pixels[k, j - 1][1] - pixels[k - 1, j][1] - pixels[k + 1, j][1] - pixels[k + 1, j + 1][1] - pixels[k, j + 1][1] - pixels[k + 1, j - 1][1] - pixels[k - 1, j + 1][1]
        blue = pixels[k, j][2] * 9 - pixels[k - 1, j - 1][2] - pixels[k, j - 1][2] - pixels[k - 1, j][2] - pixels[k + 1, j][2] - pixels[k + 1, j + 1][2] - pixels[k, j + 1][2] - pixels[k + 1, j - 1][2] - pixels[k - 1, j + 1][2]
        pixels_sharpen[k, j] = (red, green, blue, 0)

for k in range(1, im.size[0] - 1):
    for j in range(1, im.size[1] - 1):
        red = (pixels[k, j][0] + pixels[k - 1, j - 1][0] + pixels[k, j - 1][0] + pixels[k - 1, j][0] + pixels[k + 1, j][0] + pixels[k + 1, j + 1][0] + pixels[k, j + 1][0] + pixels[k + 1, j - 1][0] + pixels[k - 1, j + 1][0]) / 9
        green = (pixels[k, j][1] + pixels[k - 1, j - 1][1] + pixels[k, j - 1][1] + pixels[k - 1, j][1] + pixels[k + 1, j][1] + pixels[k + 1, j + 1][1] + pixels[k, j + 1][1] + pixels[k + 1, j - 1][1] + pixels[k - 1, j + 1][1]) / 9
        blue = (pixels[k, j][2] + pixels[k - 1, j - 1][2] + pixels[k, j - 1][2] + pixels[k - 1, j][2] + pixels[k + 1, j][2] + pixels[k + 1, j + 1][2] + pixels[k, j + 1][2] + pixels[k + 1, j - 1][2] + pixels[k - 1, j + 1][2]) / 9
        pixels_blur3x3[k, j] = (red, green, blue, 0)

for k in range(3, im.size[0] - 3):
    for j in range(3, im.size[1] - 3):
        red = green = blue = 0
        for l in range(-3, 4):
            for m in range(-3, 4):
                red = red + pixels[k + l, j + m][0]
                green = green + pixels[k + l, j + m][1]
                blue = blue + pixels[k + l, j + m][2]
        red = red / 49
        green = green / 49
        blue = blue / 49
        pixels_blur7x7[k, j] = (red, green, blue, 0)

gray1.save("Gray1_" + str(i + 1) + ".png")
gray2.save("Gray2_" + str(i + 1) + ".png")
gray3.save("Gray3_" + str(i + 1) + ".png")
sharpen.save("Sharpen_" + str(i + 1) + ".png")
blur3x3.save("Blur 3x3_" + str(i + 1) + ".png")

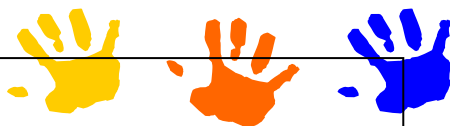
```



## LECȚIA 11

# C#, DE LA A LA Z

prof. Bibicu Dorin  
Liceul Teoretic "Dunărea" Galați



În lecția 10 am învățat :

Elemente de grafică în C#,  
clasa Graphics



În lecția 11 vom învăța :

Elemente de grafică în C#  
(continuare): chart, trimitere  
date în excel.

### 11.1 Clasa Graphics (continuare)

În lecția anterioară am prezentat clasele **System.Drawing** și **System.Drawing.Graphics** care conține metode pentru desena pe suprafețe de tip GDI+ (Graphics Device Interface). În C#, suprafețe de tip GDI+ sunt: Form, Panel, Image. Se pot desena: puncte, linii, curbe, figuri geometrice, text sau imagini.

În continuare prezentăm o aplicație pentru desena unui graf neorientat.

	x	y
	1	2
	1	4
	1	3
	2	3
	3	4

Fig. 1 Formularul "Desenare graf neorientat"



La apăsarea butonului *Citeste* din două casete text se citesc numărul de noduri  $n$  și numărul de muchii  $m$  ale grafului și se creează în mod dinamic obiectul DataGridView în care se vor introduce extremitățile  $x$  și  $y$  ale celor  $m$  muchii ale grafului.

```
private void button1_Click(object sender, EventArgs e)
{
    n = int.Parse(textBox1.Text);
    m = int.Parse(textBox2.Text);

    DataTable dt1 = new DataTable();
    dt1.Columns.Add("x", typeof(int));
    dt1.Columns.Add("y", typeof(int));

    int i;
    for (i = 1; i <= m; i++)
    {
        DataRow row = dt1.NewRow();
        dt1.Rows.Add(row);
    }

    dgv1 = new DataGridView();
    dgv1.Location = new Point(5, 150);
    dgv1.Size = new Size(250, 300);
    this.Controls.Add(dgv1);
    dgv1.DataSource = dt1;
    dgv1.AllowUserToAddRows = false;

    button2.Visible = true;
    button1.Enabled = false;
    textBox1.Enabled = false;
    textBox2.Enabled = false;
}
```



Apăsarea celui de-al doilea buton *Citeste* presupune citirea șirului se valori  $x$  și  $y$  din DataGridView, crearea matricei de adiacență a grafului, generarea în mod aleator a coordonatelor celor  $n$  noduri ale grafului, desenarea punctelor corespunzătoare nodurilor grafului, desenarea muchiilor grafului.

```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        grafic.Clear(Color.Yellow);
        grafic.Dispose();
    }
    catch { }

    int i,j,x,y;

    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++)
            a[i, j] = 0;

    for (i = 0; i <= m-1; i++)
```

```

{
    DataGridViewRow row = dgv1.Rows[i];
    x = int.Parse(row.Cells["x"].Value.ToString());
    y = int.Parse(row.Cells["y"].Value.ToString());

    a[x,y] = 1;
    a[y,x] = 1;
}

button3.Visible = true;
grafic=panel1.CreateGraphics();
mypen=new Pen(System.Drawing.Color.Red,2);

Random rd1 = new Random();

for (i = 1; i <= n; i++)
{
    vf[i, 0] = rd1.Next(0, 400);
    vf[i, 1] = rd1.Next(0, 300);

    grafic.DrawLine(mypen, vf[i, 0], vf[i, 1], vf[i, 0] + 10, vf[i, 1] + 10);
    grafic.DrawLine(mypen, vf[i, 0] + 10, vf[i, 1], vf[i, 0], vf[i, 1] + 10);

    string str = i.ToString();

    System.Drawing.Font drawFont = new System.Drawing.Font("Arial", 16);
    System.Drawing.SolidBrush drawBrush = new
    System.Drawing.SolidBrush(System.Drawing.Color.Black);
    System.Drawing.StringFormat drawFormat = new System.Drawing.StringFormat();

    grafic.DrawString(str, drawFont, drawBrush, vf[i, 0], vf[i, 1],
drawFormat);
}

mypen.Color= Color.Blue;
for(i=1;i<=n-1;i++)
    for(j=i+1;j<=n;j++)
    {
        if(a[i,j]==1)
        {
            grafic.DrawLine(mypen,vf[i, 0], vf[i, 1], vf[j, 0], vf[j, 1]);
        }
    }
}

```



În metoda **Paint** a lui **Form1**:

```

private void Form1_Paint(object sender, PaintEventArgs e)
{
    int L = this.Width;
    int l = this.Height;

    //o penita
    Pen p = new Pen(Color.Blue, 10);

```



```

        Rectangle r = new Rectangle(new Point(110, 60), new Size(L-130, 1-140));

        e.Graphics.DrawRectangle(p, r);

        String s = "Desenare cu Graphics";
        Font f = new Font("Arial", 22, FontStyle.Bold);
        SolidBrush sb = new SolidBrush(Color.Red);
        Point pt = new Point(180, 25);

        e.Graphics.DrawString(s, f, sb, pt);

        p.Dispose();
    }

```

În continuare sunt prezentate metodele Click ale celor 13 butoane ale aplicației:

```

private void button1_Click_1(object sender, EventArgs e)
{
    //linie
    Graphics g = this.CreateGraphics();

    Pen p = new Pen(Color.Red, 5);
    g.DrawLine(p, new Point(250, 150), new Point(300, 100));
}

private void button2_Click(object sender, EventArgs e)
{
    //dreptunghi
    Graphics g = this.CreateGraphics();

    Pen p = new Pen(Color.Yellow, 5);
    Rectangle r = new Rectangle(new Point(200, 100), new Size(30, 100));
    g.DrawRectangle(p,r);
}

private void button3_Click(object sender, EventArgs e)
{
    //arc de cerc
    Graphics g = this.CreateGraphics();
    Pen p = new Pen(Color.Green, 5);

    g.DrawArc(p, 350, 150, 200, 100, 0, 180);
}

private void button4_Click(object sender, EventArgs e)
{
    //elipsa
    Graphics g = this.CreateGraphics();
    Pen p = new Pen(Color.Orange, 5);

    g.DrawEllipse(p, 450, 150, 200, 100);
}

private void button5_Click(object sender, EventArgs e)
{

```



```

//poligon
Graphics g = this.CreateGraphics();
Pen p = new Pen(Color.Black, 5);
p.DashStyle = System.Drawing.Drawing2D.DashStyle.Dot;

Point[] pt = { new Point(290, 90), new Point(240, 130), new Point(250, 200), new
Point(160, 100) };

g.DrawPolygon(p, pt);
}

private void button13_Click(object sender, EventArgs e)
{
    //imagine
    Graphics g = this.CreateGraphics();
    g.DrawImage(Image.FromFile("catel.png"), new Point(600, 100));
}

private void button11_Click(object sender, EventArgs e)
{
    //text
    Graphics g = this.CreateGraphics();

    String s = "Acesta este un text";
    Font f = new Font("Batang", 22, FontStyle.Bold);
    SolidBrush sb = new SolidBrush(Color.Yellow);
    Point pt = new Point(380, 125);

    g.DrawString(s, f, sb, pt);
}

private void button6_Click(object sender, EventArgs e)
{
    //fill dreptunghi
    Graphics g = this.CreateGraphics();

    Pen p = new Pen(Color.Orange, 5);
    Rectangle r = new Rectangle(new Point(200, 100), new Size(30, 100));
    g.DrawRectangle(p, r);

    //pensula solida
    SolidBrush s = new SolidBrush(Color.Magenta);

    g.FillRectangle(s, r);
}

private void button8_Click(object sender, EventArgs e)
{
    //elipsa hasurata
    Graphics g = this.CreateGraphics();

    Pen p = new Pen(Color.Orange, 5);
    Rectangle r = new Rectangle(new Point(500, 100), new Size(30, 100));

```



```

        g.DrawEllipse(p, r);

        //pensula de hasurare
        HatchBrush s = new HatchBrush
(HatchStyle.DiagonalCross, Color.Magenta, Color.White);

        g.FillEllipse(s, r);
    }

private void button9_Click(object sender, EventArgs e)
{
    //fill gradient poligon
    Graphics g = this.CreateGraphics();
    Pen p = new Pen(Color.Black, 5);
    p.DashStyle = System.Drawing.Drawing2D.DashStyle.Dot;

    Point[] pt = { new Point(290, 90), new Point(240, 130), new Point(250, 200), new
Point(160, 100) };

    g.DrawPolygon(p, pt);

    Rectangle r = new Rectangle(new Point(290, 90), new Size(160, 150));
    LinearGradientBrush lg = new LinearGradientBrush(r, Color.Red, Co-
lor.Blue, LinearGradientMode.Horizontal);
    g.FillPolygon(lg, pt);
}

private void button10_Click(object sender, EventArgs e)
{
    //fill textura dreptunghi
    Graphics g = this.CreateGraphics();
    Pen p = new Pen(Color.Orange, 5);
    Rectangle r = new Rectangle(new Point(200, 100), new Size(230, 190));
    //g.DrawRectangle(p, r);
    //pensula solida
    TextureBrush s = new TextureBrush(Image.FromFile("catel.png"));
    g.FillRectangle(s, r);
}

private void button14_Click(object sender, EventArgs e)
{
    //rotatie dreptunghi
    Graphics g = this.CreateGraphics();
    Pen p = new Pen(Color.Yellow, 5);
    Rectangle r = new Rectangle(new Point(500, 100), new Size(30, 100));
    g.DrawRectangle(p, r);
    g.TranslateTransform(500, 100);
    g.RotateTransform(90);
    g.TranslateTransform(-500, -100);
    g.DrawRectangle(p, r);
}

private void button12_Click(object sender, EventArgs e)
{
    //Clear
    this.Invalidate();
}

```





## 11.2 Clasa Graphics (continuare): chart

În C#, clasa *Chart* permite realizarea de diagrame. Din proprietatea *Series\ChartType* se poate alege tipul de diagramă dorit.

Utilizând obiectul *chart* să se realizeze o aplicație care să permit desenarea graficului unei funcții de gradul doi.

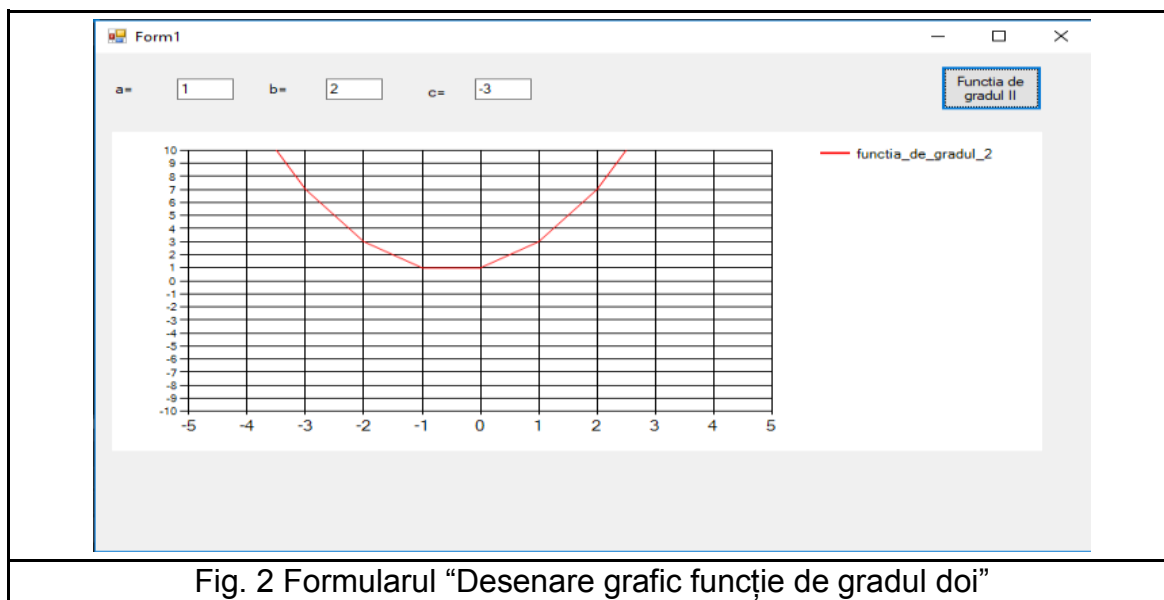


Fig. 2 Formularul “Desenare grafic funcție de gradul doi”

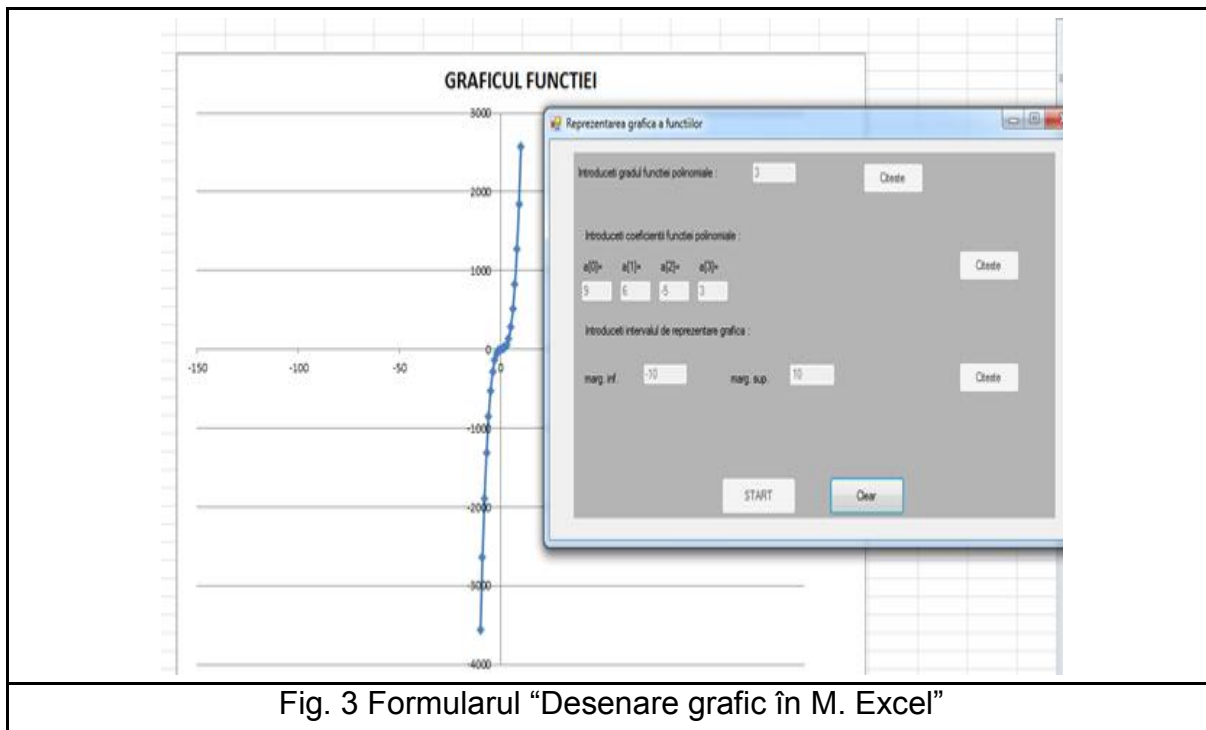
```
double f(int x)
{
    //functia
    double y = 0;
    y = a * x * x + b * x + c;
    return y;
}

private void button1_Click(object sender, EventArgs e)
{
    a = int.Parse(textBox1.Text);
    b = int.Parse(textBox1.Text);
    c = int.Parse(textBox1.Text);
    DataTable table = new DataTable();
    table.Columns.Add("x", typeof(int));
    table.Columns.Add("y", typeof(int));
    int x = 0;
    for (int i = -10; i <= 15; i++)
    {
        x = i;
        double y = 0;
        y = f(x);
        table.Rows.Add(x, y);
    }
    chart1.Series[0].Name = "functia_de_gradul_2";
    chart1.Series["functia_de_gradul_2"].Color = Color.Red;
    chart1.Series[0].Points.Clear();
    DataTableReader dr = new DataTableReader(table);
    while (dr.Read())
    {
        chart1.Series[0].Points.AddXY(dr.GetInt32(0), dr.GetInt32(1));
    }
}
```

### 11.3 Clasa Graphics (continuare): trimitere date în excel

O posibilitate pentru trasarea graficelor unor funcții o reprezintă trimiterea datelor într-un fișier M.Excel și generarea graficului cu ajutorul acestei aplicații.

Să se realizeze o aplicație care să permit desenarea graficului unei funcții polinomiale.



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;
using System.Reflection;
using System.Data.OleDb;
using System.Diagnostics;

namespace grafic1
{
    public partial class Form1 : Form
    {
        public static int n, i;
        public static TextBox[] txt=new TextBox[100];
        public static Label[] lb = new Label[100];
        public static double[] a = new double[100];
        public static int inf, sup;
        public static Process process = new Process();

        public static double mm;
        public static char[] ff = new char[100];
        public static int cont = -1;
    }
}
```

```

public Form1()
{
    InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
    n = Convert.ToInt32(textBox1.Text);
    n = int.Parse(textBox1.Text);
    label2.Visible = true;
    button2.Visible = true;
    textBox1.Enabled = false;
    button1.Enabled = false;

    for (i = 0; i <= n; i++)
    {
        string nume = "caseta_" + i.ToString("00");
        txt[i] = new TextBox();
        txt[i].Name = nume;
        txt[i].Location = new Point(i * 50+10, 120);
        txt[i].Size = new Size(40, 40);
        panel1.Controls.Add(txt[i]);

        string nume1 = "lb_" + i.ToString("00");
        lb[i] = new Label();
        lb[i].Name = nume1;
        lb[i].Location = new Point(i * 50 + 10, 100);
        lb[i].Size = new Size(40, 40);
        lb[i].Text="a["+i+"]=";
        panel1.Controls.Add(lb[i]);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    for (i = 0; i <= n; i++)
    {
        a[i] = Convert.ToDouble(txt[i].Text);
        a[i] = double.Parse(txt[i].Text);
        txt[i].Enabled = false;
    }

    label3.Visible = true;
    label4.Visible = true;
    label5.Visible = true;
    textBox2.Visible = true;
    textBox3.Visible = true;
    button3.Visible = true;
    button2.Enabled = false;
}

private void button3_Click(object sender, EventArgs e)
{
    inf = Convert.ToInt32(textBox2.Text);
    inf = int.Parse(textBox2.Text);
    sup = Convert.ToInt32(textBox3.Text);
    sup = int.Parse(textBox3.Text);
    button5.Visible = true;
    textBox2.Enabled = false;
    textBox3.Enabled = false;
    button3.Enabled = false;
}

```



```

private void button5_Click(object sender, EventArgs e)
{
    string str = @"bacalaureat.xls";
    process.StartInfo.FileName = str;
    process.Start();

    string connectionString = @"Provider=Microsoft.ACE.OleDb.15.0;Data
Source=bacalaureat.xls;Extended Properties=""Excel 8.0;HDR=YES;""";
    OleDbConnection connExcel = new OleDbConnection(connectionString);
    connExcel.Open();
    cmdExcel3.Connection = connExcel;

    for (i = inf; i <= sup; i++)
    {
        cmdExcel3.CommandText = "update [Sheet2$] set y=@ii1 where x=@ii2";
        cmdExcel3.Parameters.Add("@ii1", SqlDbType.Int);

        double m=0;

        for(int j=0;j<=n;j++)
        {
            m=m + a[j] * Math.Pow(i, j);
        }

        Console.WriteLine(m);
        cmdExcel3.Parameters["@ii1"].Value = m;
        cmdExcel3.Parameters.Add("@ii2", SqlDbType.Int);
        cmdExcel3.Parameters["@ii2"].Value = i;

        cmdExcel3.ExecuteNonQuery();
        Console.WriteLine(m);
    }

    connExcel.Close();
    button4.Visible = true;
    button5.Enabled = false;
    process.Close();
}

private void button4_Click(object sender, EventArgs e)
{
    Console.WriteLine("sal");
    string connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=bacalaureat.xls;Extended Properties=""Excel 8.0;HDR=YES;""";
    OleDbConnection connExcel = new OleDbConnection(connectionString);
    OleDbCommand cmdExcel1 = new OleDbCommand();
    OleDbCommand cmdExcel2 = new OleDbCommand();
    OleDbCommand cmdExcel3 = new OleDbCommand();

    connExcel.Open();
    cmdExcel2.Connection = connExcel;
    cmdExcel3.Connection = connExcel;

    for (i = inf; i <= sup; i++)
    {
        cmdExcel3.CommandText = "update [Sheet2$] set y=null where x=@ii2";
        cmdExcel3.Parameters.Add("@ii2", SqlDbType.Int);
        cmdExcel3.Parameters["@ii2"].Value = i;

        cmdExcel3.ExecuteNonQuery();
    }
}

```



```

for (i = 0; i <= n; i++)
{
    txt[i].Dispose();
    lb[i].Dispose();
    a[i] = 0;
}
inf = 0;
sup = 0;
n = 0;

button5.Visible = false;
button4.Visible = false;
button3.Visible = false;
button2.Visible = false;
label5.Visible = false;
label4.Visible = false;
label3.Visible = false;
label2.Visible = false;
textBox3.Visible = false;
textBox2.Visible = false;
textBox1.Enabled = true;
button1.Enabled = true;
textBox2.Enabled = true;
textBox3.Enabled = true;
button2.Enabled = true;
button3.Enabled = true;
button5.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
connExcel.Close();
    }
}

```

### **TEMĂ PROPUȘĂ**

Să se realizeze o aplicație C#, care să deseneze un graf orientat.

Să se realizeze o aplicație C#, care să deseneze utilizând obiectul chart diagrama tip coloane ce afișează distribuția mediilor unei clase de elevi.

Să se realizeze o aplicație C#, care reprezintă în M. Excel graficul funcției sinus.

### **În numărul următor al revistei vom învăța (Lecția 12) :**

12. Obiectul Timer. Aplicație – realizarea unui SlideShow automat.



**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a V-a – Problema 1—NUMERE ISTETE**  
**Prof. Stratulat Daniela – Colegiul Național "Costache Negri"**

În regatul numerelor naturale se organizează un concurs pentru desemnarea *celui mai puternic număr isteț*. Cele mai curajoase numere care se înscriu în concurs sunt obligate să se transforme în numere "istețe". Astfel, dacă numărul este par, atunci el își va păstra doar cifrele pare, iar dacă este impar își va păstra doar cifrele impare. După această transformare, toate numerele se vor alinia în fața regelui, iar acesta va alege numărul isteț cu valoarea cea mai mare. Acesta va fi desemnat câștigătorul concursului și va primi titlul de "*cel mai puternic număr isteț*".

**Cerință**

Pentru  $n$  numere naturale distincte înscrise în concurs să se determine numerele istețe în care acestea se transformă și câștigătorul titlului de "*cel mai puternic număr isteț*". În situația în care există mai multe numere istețe cu valoare maximă, regele va anunța și numărul de câștigători.

**Date de intrare**

În fișierul **istete.in**, pe primul rând este memorat un număr natural nenul  $n$  ( $n \leq 10$ ) cu semnificația din enunț, pe următorul rând sunt memorate cele  $n$  numere naturale, de maxim 9 cifre fiecare, separate prin spațiu.

**Date de ieșire**

În fișierul **istete.out** se afișează pe primul rând, separate prin spațiu, numerele "istețe" obținute, iar pe rândul următor se afișează un cel mai puternic număr isteț. Dacă există doi sau mai mulți câștigători, se afișează și numărul acestora, pe al treilea rând.

**Restricții și precizări**

- $n$  număr natural,  $1 \leq n \leq 10$ , reprezentând numărul de numere naturale distincte înscrise în concurs
- numerele citite au maxim 9 cifre fiecare

**Exemple**

istete.in	istete.out	Explicație
3 13467 2418 3456	137 248 46 248	<ul style="list-style-type: none"><li>• 13467 este număr impar și se va transforma în numărul isteț 137 (deci își va păstra doar cifrele impare în ordinea în care acestea apar în număr), transformarea are loc analog și pentru celelalte numere;</li><li>• Cel mai puternic număr isteț este 248 și apare o singură dată.</li></ul>
4 20754 123 100 92014	204 13 0 204 204 2	<ul style="list-style-type: none"><li>• Când există două sau mai multe numere istețe cu valoare maximă, se afișează și numărul de câștigători (2, în acest caz).</li></ul>

**Timp de execuție 1s/test**



## OLIMPIADA DE INFORMATICĂ

### ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018

#### Clasa a V-a – Problema 2—CIFRĂ PRIMĂ

Prof. Lăzărescu Andreea Monica – Liceul Teoretic "Mircea Eliade"

Se dă o veste mare în țara lui Mazăre Vodă. Împăratul își va mărita fata cu cel mai viteaz și isteț voinic. Din mulțimea de tineri ce aspirau la mâna domniței, doar doi dintre aceștia își făcuseră remarcate calitățile. Pentru a se decide căruia din ei îi va încredința fiica, împăratul i-a mai supus la o probă. Fiecăruia dintre ei li s-a dat un bilețel pe care era scris un număr natural de maxim 9 cifre. Voinicii trebuiau să calculeze **cifra de control** a numărului precizat (să însumeze cifrele numărului inițial, apoi să efectueze suma cifrelor numărului rezultat, procedeul repetându-se până la obținerea unei cifre, cifră ce reprezintă cifra de control a numărului inițial). Dacă vreun candidat obține o cifră număr prim, atunci el se va căsători cu fiica de împărat; dacă ambii tineri obțin în urma calculului cifre prime, câștigătorul va fi cel ce a primit inițial un număr mai mic pe bilet; dacă niciun candidat nu obține o cifră primă, căsătoria nu va mai avea loc.

#### Cerință

Scrieți un program care citește cele două numere naturale nenule **x** și **y** de pe bilețele și care afișează **x**, dacă doar cifra de control a lui **x** e număr prim, **y**, dacă doar cifra de control a lui **y** e primă, minimul dintre **x** și **y**, dacă ambele numere au cifra de control primă și 0 dacă nu se obține o cifră primă.

#### Date de intrare

Pe prima linie în fișierul **cifra.in** sunt numerele naturale **x** și **y**, separate printr-un spațiu.

#### Date de ieșire

Pe prima linie în fișierul **cifra.out** se va scrie **x**, dacă doar cifra de control a lui **x** e număr prim, **y**, dacă doar cifra de control a lui **y** e primă, minimul dintre **x** și **y**, dacă ambele numere au cifra de control primă sau 0 (dacă niciun număr nu are o cifră de control primă).

#### Restricții și precizări

- Numerele de pe bilețele sunt numere naturale de maxim 9 cifre și minim 2 cifre
- Dacă ambele numerele nu induc o cifră primă, se va afișa 0 pe prima linie din fișier.
- Cele două numere sunt distincte.

#### Exemple

cifra.in	cifra.out	Explicație
12 109878909	12	<ul style="list-style-type: none"><li>• 12 are cifra de control 3 - număr prim, iar 109878909 are cifra de control 6 - neprim. Prin urmare se va afișa 12.</li></ul> <p>Cifra de control a unui număr (de ex. 109878909) se calculează astfel: se însumează cifrele (<math>1+0+9+8+7+8+9+0+9=51</math>); dacă se obține un număr cu mai mult de o cifră, se face suma cifrelor numărului obținut (<math>5+1=6</math>). Algoritmul continuă până când se obține un număr cu o singură cifră. Aceea este cifra de control.</p>
1533 9893	1533	<ul style="list-style-type: none"><li>• Ambele numere au cifra de control 3 – număr prim (<math>1533 \rightarrow 3</math>, iar <math>9893 \rightarrow 2</math>), dar numărul mai mic este 1533.</li></ul>
148 999	0	<ul style="list-style-type: none"><li>• 148 are cifra de control 4, iar 999 are cifra de control 9. Nici un număr nu este prim, deci se afișează 0.</li></ul>

Timp de execuție 1s/test



**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a VI-a – Problema 1—EDUCAȚIE FIZICĂ**  
**Prof. Ilie Daniela – Colegiul Național "Al.I.Cuza" Galați**

La ora de educație fizică elevii primesc câte un tricou cu logo-ul școlii și un număr. Doamna profesoară le cere elevilor să meargă pe terenul de sport și să se alinieze în ordinea crescătoare a numerelor de pe tricouri. Când doamna ajunge pe teren, constată că sunt doar  $n$  elevi aliniați în ordinea numerelor de pe tricouri, iar restul de  $m$  elevi întârzie. Când vor ajunge și cei  $m$  elevi, aceștia trebuie să se așeze printre elevii deja aliniați, astfel încât toți cei  $m+n$  elevi să fie aliniați în ordinea crescătoare a numerelor de pe tricouri. Cei  $m$  elevi nu ajung toți odată pe teren. Când apare un elev, doamna trebuie să-i găsească locul și să-l așeze în șirul de elevi, păstrând ordinea crescătoare a numerelor de pe tricouri.

**Cerință**

Cunoscându-se valorile  $n$  și  $m$  precum și faptul că elevii care au întârziat se vor alinia pe rând în ordinea sosirii lor pe terenul de sport, astfel încât ordinea crescătoare să se păstreze, scrieți un program care determină pozițiile pe care trebuie așezați elevii care au întârziat atunci când sosesc. Programul va afișa și șirul celor  $n+m$  valori ale numerelor de pe tricouri la final.

**Date de intrare**

În fișierul **ora.in** pe prima linie sunt memorate numerele naturale nenule  $n, m$  ( $n \leq 50, m \leq 50$ ), pe a doua linie, sunt memorate  $n$  numere naturale nenule ordonate strict crescător și pe linia a treia sunt memorate  $m$  numere naturale nenule, cu semnificația din enunț. *Numerele de pe tricouri sunt distincte (nu pot exista mai mulți elevi cu același număr pe tricou).*

**Date de ieșire**

În fișierul **ora.out** se scriu pe primul rând  $m$  valori reprezentând pozițiile pe care s-au așezat elevii care au întârziat atunci când au ajuns pe teren. Pe rândul al doilea se scriu valorile numerelor de pe tricouri în ordinea în care sunt aliniați elevii la final, după sosirea tuturor elevilor la oră.

**Restricții și precizări**

- $n, m$  ( $n \leq 50, m \leq 50$ )
- numerele de pe tricourile elevilor sunt numere naturale nenule de cel mult 4 cifre

**Exemple**

ora.in	ora.out	Explicație
6 4 7 10 13 21 43 52 22 6 99 12	5 1 9 4 6 7 10 12 13 21 22 43 52 99	6- elevi existenți pe teren 7 10 13 21 43 52-numerele de pe tricou 4- numarul elevilor care intarzie 22 6 99 12- numerele de pe tricou 5- pozitia pe care se așează elevul cu numarul 22 atunci când apare 1- pozitia pentru elevul cu numarul 6 pe tricou 9- pozitia pentru elevul cu numarul 99 pe tricou 4- pozitia pentru elevul cu numarul 12 pe tricou 6 7 10 12 13 21 22 43 52 99-așezarea la final
4 3 45 51 74 86 94 47 78	5 2 5 45 47 51 74 78 86 94	5- pozitia pe care se așează elevul cu numarul 94 atunci când apare 2- pozitia pe care se așează elevul cu numarul 47 atunci când apare 5- pozitia pe care se așează elevul cu numarul 78 atunci când apare 45 47 51 74 78 86 94 - așezarea la final

**Timpe de execuție 2s/test**





**OLIMPIADA DE INFORMATICĂ  
ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**

**Clasa a VI-a – Problema 2—JOC**

**Prof. Cobzaru Luminița – Colegiul Național "Vasile Alecsandri" Galați**

Albert și Philip sunt frați. Albert este mezinul și dorește să joace și el jocurile online cu care se distrează fratele său. Pentru a-și proteja strategia, Philip și-a parolat jocurile cu ajutorul unor cuvinte formate doar din litere mari, dar Albert știe că parolele trebuie să fie formate doar din cifre. Albert își găsește un aliat, mama lor, pe care o roagă să afle algoritmul pe care îl aplică Philip. Mama reușește să afle că fiecare literă din cuvânt este transformată într-un număr, număr ce reprezintă codul ASCII al literei (mama nu știe ce înseamnă codul ASCII și nici nu știe dacă a pronunțat corect). Aceste numere sunt adunate, iar cu toate cifrele care formează această sumă, Philip construiește cel mai mic număr. Acesta ar fi parola jocului.

**Cerință**

Cunoscându-se cuvântul lui Philip, scrieți un program care să determine cel mai mic număr obținut cu toate cifrele sumei codurilor ASCII din cuvântul dat.

**Date de intrare**

În fișierul **joc.in** este memorat cuvântul dat.

**Date de ieșire**

În fișierul **joc.out** se scrie numărul obținut.

**Restricții și precizări**

- cuvântul are o lungime mai mică de 250 caractere.
- cuvântul conține doar litere mari
- pentru 50% dintre teste, lungimea cuvântului e mai mică de 50
- pentru 50% dintre teste, lungimea cuvântului este între 50 și 250

**Exemple**

joc.in	joc.out	Explicație
ABAB	226	Suma codurilor ASCII este $65(A) + 66(B) + 65(A) + 66(B) = 262$ . Cel mai mic număr care se poate forma cu aceste cifre este 226
ESTIBUN	358	Suma codurilor ASCII este $69(E) + 83(S) + 84(T) +$ $73(I) + 66(B) + 85(U) + 78(N) = 538$ Cel mai mic număr care se poate forma cu aceste cifre este 358

**Timp de execuție 1s/test**



**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a VII-a – Problema 1—JETOANE**  
**Prof. Tudor Veronica – Liceul Teoretic "Dunărea" Galați**

Geo și Mircea, doi prieteni buni, au inventat un joc cu jetoane numit **GpriM**. Pasionați de Informatică, ei hotărăsc să scrie un program care să calculeze scorul fiecărui jucător și la final să afișeze câștigătorul. Jocul dispune de  $n$  jetoane, pe fiecare jeton fiind scris un număr natural cu maxim 4 cifre. Cei doi jucători extrag succesiv câte un jeton, citesc numărul, și dacă numărul de pe jeton este un număr prim, îl adaugă la punctajul lui, dacă nu, cu ajutorul unui creion magic poate șterge prima sau ultima cifră a numărului de pe jetonul extras și verifică dacă numerele astfel obținute sunt prime.

Dacă ambele numere obținute prin folosirea creionului magic sunt prime, jucătorul are voie să adauge la scor doar pe cel mai mic număr dintre cele două. Dacă doar unul dintre aceste două numere obținute prin ștergerea unei cifre este număr prim, acel număr va fi adăugat la scor.

În cazul în care nu găsește un număr prim pe jeton și nici nu descoperă un număr prim cu ajutorul creionului magic, scorul jucătorului rămâne nemodificat. Jocul se încheie la terminarea jetoanelor.

**Cerință**

a) Dacă cerința este 1, să se afișeze jucătorul care a câștigat jocul și punctajul obținut de acesta. Dacă cei doi jucători au obținut punctaj egal, se va scrie numele ambilor jucători. Geo este primul jucător, iar Mircea este cel de-al doilea jucător.

b) Dacă cerința este 2, să se afișeze numărul maxim de jetoane folosite de un jucător. Prin jeton folosit se înțelege un jeton cu ajutorul căruia un jucător își mărește punctajul.

**Date de intrare**

De pe prima linie a fișierului de intrare **jeton.in** se citește un număr natural  $c$  (care poate lua valorile 1 sau 2), reprezentând cerința de rezolvat. De pe a doua linie a fișierului de intrare se citește un număr natural  $n$ , reprezentând numărul de jetoane, iar de pe linia următoare se citesc  $n$  numere naturale nenule cu cel mult 4 cifre fiecare. Numerele sunt separate printr-un spațiu.

**Date de ieșire**

Dacă cerința este 1, în fișierul de ieșire **jeton.out** se scrie pe prima linie numele câștigătorului separat printr-un spațiu de scorul obținut de el. Dacă ambii jucători au obținut același scor, în fișier se scriu numele ambilor jucători, în ordinea **Geo Mircea**, urmate de scorul obținut de cei doi. Dacă cerința este 2, se scrie pe prima linie a fișierului **jeton.out** numărul maxim de jetoane folosite de un jucător la mărirea punctajului său, fie el câștigător sau nu.

**Restricții și precizări**

- $n$ , număr natural nenul,  $1 \leq n \leq 2000000$ .
- $c$  număr natural, ce poate lua valorile 1 sau 2.

**Timp maxim de execuție pe test 2.5 s/test**

**Memorie totală disponibilă: 1 MB.**

**Dimensiunea maximă a sursei 1 KB**

**Exemple**

jeton.in	jeton.out	Explicație
1 6 17 234 15 82 64 326	Mircea 25	<b>Pentru cerința 1 și <math>n=6</math></b> Geo extrage jetoanele: 17, 15 și 64. Punctajul lui se obține astfel: $17+5$ (obținut din 15 prin ștergerea primei cifre)=22 Mircea extrage jetoanele: 234, 82 și 326. Punctajul lui este: 23 (obținut din 234 prin ștergerea ultimei cifre)+2 (obținut din 82)=25 Pentru cerința 1 se va afișa: Mircea 25
2 6 17 234 15 82 64 326	2	<b>Cerința este 2.</b> Conform explicației de mai sus, atât Geo cât și Mircea folosesc câte 2 jetoane, deci 2 este numărul maxim de jetoane pe care le adaugă un jucător.
1 9 91 21 237 13 156 342 28 105 196	Geo 44	<b>Cerința este 1 <math>n=9</math></b> Geo extrage jetoanele: 91, 237, 156, 28 și 196. Punctajul lui se calculează astfel: $23+2+19=44$ 237 nu e prim, dar 23 și 37 da. Minimul e 23. Mircea extrage jetoanele: 21, 13, 342 și 105. Punctajul lui este: $2+13+5=20$ Pentru cerința 1 se va afișa: Geo 44 (deoarece Geo are punctajul cel mai mare)
2 9 91 21 237 13 156 342 28 105 196	3	<b>Cerința este 2.</b> Geo calculează $23+2+19=44$ , deci folosește 3 jetoane și Mircea $2+13+5=20$ folosește 3 jetoane, deci 3 este numărul maxim de jetoane pe care le adaugă un jucător.
1 6 91 21 237 237 21 91	Geo Mircea 25	Geo extrage jetoanele: 91, 237, 21. Punctajul lui se calculează astfel: $23+2=25$ Mircea extrage jetoanele: 21, 237 și 91. Punctajul lui este: $2+23=25$ Cei doi au obținut punctaje egale.

## OLIMPIADA DE INFORMATICĂ ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018

### Clasa a VII-a – Problema 2—COD

Prof. Georgeta Balacea – Colegiul Național "Vasile Alecsandri" Galați

Daphne, Vrăjitoarea Vânturilor din Tărâmul Fantasy, a pus la grea încercare pe copiii familiei Walker. Aceștia își vor putea recupera părinții răpiți de Daphne doar dacă vor găsi Cartea Secretelor. Brandon, cel mai mic dintre frații Walker, descoperă că pentru a recupera cartea cerută de vrăjitoare trebuie să găsească un COD iar pentru a o putea deschide, trebuie să descopere CODUL SECRET. CODUL este format din  $2 \cdot n$  numere cu cel mult nouă cifre, ascunse printre elementele unui tablou bidimensional cu  $n$  linii și  $n$  coloane. Primele  $n$  numere naturale din care este compus codul se găsesc în ordine, de sus în jos, pe diagonala principală a tabloului bidimensional cu  $n$  linii și  $n$  coloane iar următoarele  $n$  numere se află în ordine, de sus în jos, pe diagonala secundară a matricei.

După ce Brandon reușește să descopere CODUL pentru a găsi Cartea Secretelor, copilul trebuie să găsească și CODUL SECRET, pentru a putea să deschidă cartea. Abia după aceea domnul și doamna Walker vor fi aduși înapoi, la copiii lor.

CODUL SECRET este format din numerele aflate în cea mai lungă secvență strict crescătoare din CODUL găsit anterior. Dacă în șirul de numere care formează CODUL se găsesc mai multe secvențe strict crescătoare, cu aceeași lungime maximă, atunci CODUL SECRET va fi format din toate numerele ce formează aceste secvențe, în ordinea în care ele apar în CODUL SECRET, de la stânga la dreapta.

O secvență este formată din elemente ale unui șir de numere, consecutive ca poziție în șir.

O secvență strict crescătoare are proprietatea că fiecare element al secvenței, exceptând primul, este strict mai mare decât elementul anterior.

*De exemplu*, pentru șirul de  $m$  numere  $a_1, a_2, \dots, a_{m-1}, a_m$  secvența  $a_i, a_{i+1}, \dots, a_{k-1}, a_k$  este strict crescătoare dacă  $1 \leq i \leq k \leq m$  și  $a_j < a_{j+1}$ , oricare ar fi  $j$ ,  $i \leq j < k$ . Lungimea acestei secvențe este  $k-i+1$ .

*De exemplu*, pentru șirul de 10 numere 9,8,4,5,7,8, 8,10,12,11 lungimea maximă a unei secvențe strict crescătoare este 4 (secvența este formată din numerele  $4 < 5 < 7 < 8$ ). Poziția lui 4 în șir este 3 iar poziția lui 8 în șir este 6. Prin urmare, lungimea secvenței este  $6-3+1=4$  (dacă se consideră că elementele din șir sunt numerotate de la 1).

*De exemplu*, pentru șirul de 10 numere 9,8,4,5,7,8, 8,10,12,11 lungimea maximă a unei secvențe strict crescătoare este 4 (secvența este formată din numerele  $4 < 5 < 7 < 8$ ). Poziția lui 4 în șir este 3 iar poziția lui 8 în șir este 6. Prin urmare, lungimea secvenței este  $6-3+1=4$  (dacă se consideră că elementele din șir sunt numerotate de la 1).

### Cerință

a) Dându-se  $2 \cdot n$  numere naturale nenule, cu cel mult nouă cifre fiecare, să se determine cele  $2 \cdot n$  numere din care este format CODUL (elementele tabloului bidimensional cu  $n$  linii și  $n$  coloane, obținute parcurgând în ordine, diagonala principală (de sus în jos) și apoi diagonala secundară (de sus în jos)).

b) Din șirul astfel obținut, să se determine elementele secvențelor strict crescătoare (din șirul obținut la punctul (a)), de lungime maximă, scrise în ordine, de la stânga la dreapta.



### Date de intrare

Din fișierul **cod.in** se citesc de pe prima linie două numere naturale **c** și **n**, unde **c** reprezintă cerința (**c=1** pentru rezolvarea cerinței (**a**) - determinarea elementelor matricei obținute prin parcurgerea diagonalei principale și apoi a diagonalei secundare, respectiv **c=2** pentru rezolvarea cerinței (**b**) – determinarea elementelor celor mai lungi secvențe strict crescătoare).

De pe fiecare dintre următoarele **n** linii ale fișierului **cod.in** se citesc câte **n** numere naturale.

### Date de ieșire

În fișierul **cod.out** se scriu pe prima linie cele **2\*n** elemente ale șirului ce reprezintă **CODUL**, dacă cerința este 1, respectiv elementele **CODULUI SECRET**, dacă cerința este 2.

### Restricții și precizări

- **c** număr natural, ce poate lua valorile 1 sau 2.
- **n**, număr natural nenul,  $1 \leq n \leq 2000$ .
- Elementele tabloului bidimensional cu **n** linii și **n** coloane sunt numere naturale nenule cu cel mult 9 cifre fiecare.
- Pentru 40% dintre teste cerința este 1.
- Pentru 50% dintre teste  $n \leq 200$ .

### Exemple

cod.in	cod.out	Explicație
1 5 93 144 134 98 164 96 146 28 100 61 39 87 101 13 36 69 135 68 175 137 125 136 196 57 107	93 146 101 175 107 164 100 101 135 125	Pentru cerința 1 și $n=5$ se cer elementele tabloului bidimensional cu 5 linii și 5 coloane, obținute prin parcurgerea diagonalei principale și a diagonalei secundare
2 5 93 144 134 98 164 96 146 28 100 61 39 87 101 13 36 69 135 68 175 137 125 136 196 57 107	100 101 135	Cerința este 2. În șirul 93 146 101 175 107 164 100 101 135 125, lungimea maximă a unei secvențe strict crescătoare este 3. Șirul celor $2*n$ (10) elemente conține o singură secvență strict crescătoare de lungime maximă (3). Prin urmare, CODUL SECRET este 100 101 135
1 6 185 93 204 12 23 87 240 99 227 171 106 56 14 40 109 120 243 91 182 223 173 116 191 188 10 29 81 107 190 82 27 175 74 68 97 94	185 99 109 116 190 94 87 106 120 173 29 27	Pentru cerința 1 și $n=6$ se cer elementele tabloului bidimensional cu 6 linii și 6 coloane, obținute prin parcurgerea diagonalei principale și a diagonalei secundare
2 6 185 93 204 12 23 87 240 99 227 171 106 56 14 40 109 120 243 91 182 223 173 116 191 188 10 29 81 107 190 82 27 175 74 68 97 94	99 109 116 190 87 106 120 173	Cerința este 2. Lungimea maximă a unei secvențe strict crescătoare este 4. Șirul obținut la punctul a (prin parcurgerea diagonalei principale și apoi a diagonalei secundare) conține două secvențe strict crescătoare de lungime 4. 185 99 109 116 190 94 87 106 120 173 29 27. Prin urmare, CODUL SECRET este 99 109 116 190 87 106 120 173
2 4 19 10 23 91 12 23 93 16 10 29 18 17 87 75 74 68	18 68 91 93	CODUL este 19 23 18 68 91 93 29 87. Lungimea maximă a unei secvențe strict crescătoare este 4. Există o singură secvență strict crescătoare de lungime 4. CODUL SECRET este 18 68 91 93.

Timp maxim de execuție pe test 2,5 s/test

Memorie totală disponibilă: 3 MB.

Dimensiunea maximă a sursei 1 KB





**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a IX-a – Problema 1—NUMĂR ETALON**  
**Profesor Popescu Mădălina Colegiul Național „Mihail Kogălniceanu” Galați**

Mihai este un elev pasionat de teoria numerelor naturale. Acest domeniu este fascinant și deschis în egală măsură atât matematicienilor de carieră cât și amatorilor. Astfel, Mihai studiind numerele a observat că pentru un număr natural **nr** poată să găsească sau nu dintr-o listă de numere un număr care are se află într-o anumită relație cu numărul nr, număr pe care Mihai l-a numit număr etalon.

Iată cum definește Mihai numărul **x** număr etalon al lui **nr**:

- Numărul **x** nu are nicio cifră în comun cu numărul **nr**
- Cele două numere, împreună au toate cifrele de la 0 la 9
- În cele două numere, numărul total de apariții al cifrelor pare să fie egal cu numărul total de apariții al cifrelor impare.

**Cerință**

Scrieți un program care să citească din fișier un număr natural **nr** și apoi mai multe numere naturale dintre care să se determine cel mai mare dintre numerele care sunt etalon cu **nr**. În cazul în care nu există un astfel de număr să se afișeze mesajul "Nu există"

**Date de intrare**

Fișierul **etalon.in** conține;

- pe prima linie un număr natural **nr**
- pe următoarele linii numere naturale, pe fiecare linie sunt numere separate prin câte un spațiu.

**Date de ieșire**

Fișierul **etalon.out** va conține pe prima linie numărul **x**, cel mai mare număr etalon din lista de numere sau în cazul în care nu există mesajul "Nu există".

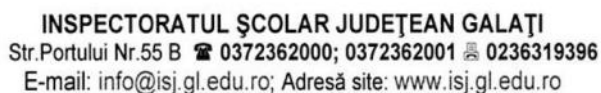
**Restricții și precizări**

- Toate numerele sunt numere naturale din intervalul **[1,1018)**
- Există cel mult 10000 de numere în fișier.

**Exemple**

**Timp maxim de execuție/test:** 0,5 secunde

etalon.in	etalon.out	Observații
123123123 346581 456789048 456 10 9 456789000	456789048	Există două numere etalon cu 123123123 : 456789048 și 456789000. Se scrie cel mai mare.
3257 41068 108 6416089	Nu exista	Niciun numar din lista nu indeplineste condi- țiile numărului etalon pentru 3257





**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a X-a – Problema 1—K-POLIPRIME**

**Prof. Neagu Violeta – Colegiul Național "Mihail Kogălniceanu"**

În orice interval de numere naturale  $[a, b]$  cu  $a < b$  și  $a > 2$  există măcar un număr natural care se poate exprima ca produs de cel puțin două numere prime (nu obligatoriu distincte). Numim *k-poliprime*, acele numere din intervalul  $[a, b]$  care se pot exprima ca un produs format dintr-un număr maxim de factori care reprezintă numere prime iar numărul de factori cu valoare maximă este  $k$ . De exemplu, în intervalul  $[15, 30]$  sunt două numere *k-poliprime*: 16 și 24, unde  $16 = 2 \times 2 \times 2 \times 2$  iar  $24 = 2 \times 2 \times 2 \times 3$  iar  $k = 4$ . Toate celelalte numere din intervalul  $[15, 30]$  care se pot exprima ca produse de numere prime au mai puțin de 4 factori, deci nu sunt *k-poliprime*.

**Cerință**

Dându-se un interval de numere naturale  $[a, b]$  cu  $a < b$  și  $a > 2$  să se determine trei numere naturale **np**, **k** și **val** unde : **np** reprezintă câte numere *k-poliprime* există în acel interval, **k** are semnificația din enunț iar **val** reprezintă un număr *k-poliprim* din acel interval. În cazul în care intervalul  $[a, b]$  conține mai multe numere *k-poliprime* se va afișa primul dintre ele care are și suma tuturor divizorilor lui cu valoare mai mare.

**Date de intrare**

În fișierul **poliprime.in** la începutul fișierului pe primele două rânduri sunt memorate două numere naturale  $[a, b]$  cu  $a < b$ .

**Date de ieșire**

În fișierul **poliprime.out** se vor scrie pe primul rând separate printr-un singur spațiu valorile lui **np**, **k** și **val**.

**Restricții și precizări**

- $2 < a < b < 10\,000\,000$
- $1 \leq b-a \leq 10000$
- Pentru 40% dintre teste  $a < b \leq 100000$
- Pentru 30% dintre teste  $a < b \leq 1000000$
- Pentru 20% dintre teste  $a < b \leq 10000000$

**Punctaj**

- Dacă ați determinat corect **np**, obțineți 30% din punctaj.
- Dacă ați determinat corect **k**, obțineți 40% din punctaj
- Dacă ați determinat corect **val**, obțineți 30% din punctaj.

**Exemple**

poliprime.in	poliprime.out	Explicație
31 60	2 5 48	În intervalul $[31, 60]$ sunt două numere <i>k-poliprime</i> : 32 și 48, unde $32 = 2 \times 2 \times 2 \times 2 \times 2$ iar $48 = 2 \times 2 \times 2 \times 2 \times 3$ . Toate celelalte numere din intervalul $[31, 60]$ care se pot exprima ca produse de numere prime au mai puțin de 5 factori, deci nu sunt <i>k-poliprime</i> . 5 reprezintă numărul de numere prime a căror produs este egal cu numărul <i>k-poliprim</i> . Există două numere <i>k-poliprime</i> în interval dar suma divizorilor lui 48 este mai mare decât suma divizorilor lui 32 Suma divizorilor lui 32: $1+2+4+8+16+32=63$ Suma divizorilor lui 48: $1+2+3+4+6+8+12+16+24+48=124 \Rightarrow$ se afișează 48 care are suma divizorilor mai mare
99 107	2 4 100	În intervalul $[99, 107]$ sunt două numere <i>k-poliprime</i> : 100 și 104, unde $100 = 2 \times 2 \times 5 \times 5$ iar $104 = 2 \times 2 \times 2 \times 13$ . Toate celelalte numere din intervalul $[99, 107]$ care se pot exprima ca produse de numere prime au mai puțin de 4 factori, deci nu sunt <i>k-poliprime</i> . 4 reprezintă numărul de numere prime a căror produs este egal cu numărul <i>k-poliprim</i> . Există două numere <i>k-poliprime</i> în interval dar suma divizorilor lui 100 este mai mare decât suma divizorilor lui 104 Suma divizorilor lui 100: $1+2+4+5+10+20+25+50+100=217$ Suma divizorilor lui 104: $1+2+4+8+13+26+52+104=210 \Rightarrow$ se afișează 100 care are suma divizorilor mai mare

**Timp maxim de execuție/test: 0,5 secunde**



**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a X-a – Problema 2—CUVÂNTUL SECRET**  
**Prof. Voinea Mirela – Colegiul Național "Vasile Alecsandri"**

Iliuță, în continuare mare cinefil, e pasionat, mai nou, de filmele cu mistere și coduri secrete. Cu o imaginație bogată, își imaginează fel de fel de scenarii. Dornic să-și perfecționeze abilitățile de spărgător de coduri, a inventat un joc pe care-l propune prietenilor lui. Într-un careu de  $nxn$  caractere, undeva, pe verticală, citit de sus în jos, există un *cuvânt secret*. Acesta este cuprins între două caractere \*. Provocarea lui Iliuță pentru prietenii săi este ca aceștia să „vadă” liniile din careu care conțin *cuvântul secret*, ascuns printre alte caractere. Cu alte cuvinte, să determine acele linii din careu care, prin ștergerea unor caractere, se pot transforma în *cuvântul secret*.

**Cerință**

Dându-se dimensiunea  $n$  a careului și un careu de  $nxn$  litere mici din alfabetul englez printre care se află, pe aceeași coloană, două caractere \*, să se determine numărul de linii care, prin ștergerea unor caractere, se pot transforma în *cuvântul secret*.

**Date de intrare**

În fișierul **secret.in** este memorat pe primul rând un număr natural nenul  $n$  ( $4 \leq n \leq 500$ ) cu semnificația din enunț, iar pe următoarele  $n$  rânduri sunt câte  $n$  caractere, litere mici ale alfabetului englez, cu excepția a două caractere \*, aflate pe aceeași coloană.

**Date de ieșire**

În fișierul **secret.out** se scrie pe primul rând numărul de linii din careu care se pot transforma în *cuvântul secret*.

**Restricții și precizări**

- $n$  număr natural,  $4 \leq n \leq 500$
- Există în careu exact două caractere \*, situate pe aceeași coloană. Între ele se află *cuvântul secret*, care are o lungime de minim 2 caractere.
- Pentru 40% dintre teste  $n \leq 50$

**Exemple**

secret.in	secret.out	Explicație
5 acbb* acbca alcnb babac baca*	2	2 reprezintă numărul de linii care, prin ștergerea unor caractere, se pot transforma în <i>cuvântul secret</i> . Cuvântul secret este <b>abc</b> . Liniile sunt: <b>acbca</b> , <b>babac</b> (literele îngroșate formează <i>cuvântul</i> , celelalte se șterg)
7 afianeo if*onfo foi- nifo fonwild infefou iuoanxf io*fstn	3	3 reprezintă numărul de linii care, prin ștergerea unor caractere, se pot transforma în <i>cuvântul secret</i> . Cuvântul secret este <b>info</b> . Liniile sunt: <b>if*onfo</b> , <b>foinifo</b> , <b>infefou</b>
4 aa*b aabc aabd aa*b	0	Cuvântul secret este <b>bb</b> . Nicio linie nu se poate transforma în <b>bb</b> .

**Timp maxim de execuție/test: 0,5 secunde**

**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a XI-XII -a – Problema 1—FLORI**  
**Prof. Genia Joc – Colegiul Național "Al. I. Cuza"**

O grădină, privită de sus, are formă dreptunghiulară și este compusă din  $n$  rânduri identice, iar pe fiecare rând sunt  $m$  parcele care au formă de pătrat. Aceasta este înconjurată de un gard care are exact două porți. Fiecare parcelă conține un sortiment de flori codificat printr-un număr natural.

Violeta, o fetiță care iubește foarte mult florile, dorește să parcurgă grădina de la o poartă la alta. Ea se poate deplasa doar paralel cu laturile dreptunghiului (direcțiile est-vest-nord-sud) și poate avansa doar o parcelă (celulă) la un moment dat, iar la florile din anumite parcele face alergie (dacă numărul de cifre 0 este egal cu numărul de cifre 1 din reprezentarea binară a codului parcelei).

**Cerință**

Cunoscând  $n$  și  $m$  dimensiunile grădinii, codurile florilor din fiecare dintre cele  $n \cdot m$  parcele, coordonatele celor două porți, determinați:

- Numărul  $X$  de parcele la care Violeta face alergie, dacă reprezentarea binară a codului parcelei are numărul de cifre 0 egal cu numărul de cifre 1.
- Numărul minim de parcele pe care le parcurge fetița pentru a traversa grădina de la o poartă la alta.

**Date de intrare**

Fișierul de intrare **flori.in** conține:

- pe prima linie două numere naturale  $n$  și  $m$  separate prin câte un spațiu, cu semnificațiile din enunț;
- pe următoarele  $n$  linii câte  $m$  numere naturale, separate prin câte un spațiu, reprezentând codurile florilor de la cele  $n \cdot m$  parcele ale grădinii.
- ultima linie a fișierului conține patru numere naturale  $X1$ ,  $Y1$ ,  $X2$ ,  $Y2$ , separate prin câte un spațiu, reprezentând pozițiile celor două porți ( $X1$ ,  $Y1$  reprezintă linia și respectiv coloana zonei ce conține prima poartă, iar  $X2$ ,  $Y2$  reprezintă linia și respectiv coloana zonei ce conține cea de-a doua poartă)

**Date de ieșire**

În fișierul de ieșire **flori.out** se va scrie

- pe primul rând numărul natural  $X$ , reprezentând numărul total de parcele la care fetița face alergie
- pe al doilea rând numărul minim de parcele pe care le parcurge fetița pentru a traversa grădina de la o poartă la alta.

**Restricții și precizări**

- $2 \leq n, m \leq 150$ ,  $1 \leq \text{codul} \leq 9999$
- Pentru toate testele problemei există soluție, adică fetița poate ajunge de la o poartă la alta.
- Pentru rezolvarea corectă a primei cerințe se acordă **20** de puncte, iar pentru cerința a doua se acordă **80** de puncte.
- Pentru 70% dintre teste, se garantează  $m \leq 50$  și  $n \leq 50$ .

**Exemple**

flori.in	flori.out	Explicație
4 5 8 7 3 2 9 8 5 6 2 9 2 4 1 6 8 2 7 8 4 6 1 1 4 5	6 8	Dacă notăm cu 1 parcelele accesibile și cu 0 cele inaccesibile, obținem următoarea matrice: 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 1 Doar numerele 2 și 9 au numărul cifrelor de 0 egal cu numărul cifrelor de 1 în reprezentarea binară. Sunt 6( $X=6$ ) parcele în care fetița nu are acces. Violeta se află în parcela de coordonate (1,1) și trebuie să ajungă în parcela de coordonate (4,5); unul din drumurile de lungime minimă este ; (1,1)→(1,2)→(2,2)→(2,3)→(3,3)→(3,4)→(3,5)→(4,5)

Timp maxim de execuție/test: 1 s



**OLIMPIADA DE INFORMATICĂ**  
**ETAPA LOCALĂ, GALAȚI, 25 FEBRUARIE 2018**  
**Clasa a XI-XII -a – Problema 2—SPONSOR**  
**Prof. Magdalena Grigore – Colegiul Național "Al.I.Cuza"**

La un concurs de talente s-au înscris mai mulți artiști. Fiecare artist a primit la înscriere un număr de ordine, în ordinea sosirii. În funcție de prestație, fiecărui concurent i s-au acordat punctaje și cele mai mari trei punctaje au primit premiu. Un sponsor care se pricepea la artă, dar marea lui pasiune era algoritmica, a dorit să acorde premii suplimentare unor concurenți, dar ca să nu aleagă neapărat pe cei mai buni și să nu impună o anumită sumă, sponsorul a vrut a creat următorul algoritm, spunându-și:

- ◇ Ordonez alfabetic concurenții
- ◇ După ordonare voi obține șirul numerelor de ordine modificat
- ◇ Voi acorda premii concurenților cu numerele de ordine care fac parte din cel mai lung subșir crescător, comun șirului format cu numerele de ordine inițiale și celui obținut din numerele de ordine după ordonarea alfabetică în funcție de numele concurenților.
- ◇ Dacă voi obține mai multe astfel de șiruri, de aceeași lungime maximă, voi alege primul șir.
- ◇ Dacă subșirul comun are cel mult 3 valori, valoarea premiului va fi lungimea maximă obținută, altfel valoarea premiului va fi egală cu restul împărțirii dintre suma valorilor din subșirul comun și lungimea maximă.

**Cerință**

Dându-se un număr natural  $n$  și apoi un șir de câte  $n$  șiruri cu maximum 15 caractere litere ale alfabetului englez ce reprezintă numele concurenților, să se afișeze valoarea premiului acordat suplimentar de către sponsor.

**Date de intrare**

În fișierul **sponsor.in** pe prima linie a fișierului este memorat un număr natural nenul  $n$  ( $n \leq 100$ ), urmat de  $n$  șiruri de caractere cu proprietatea din cerință.

**Date de ieșire**

În fișierul **sponsor.out** se scrie pe primul rând valoarea premiului.

**Restricții și precizări**

sponsor.in	sponsor.out	Explicație																				
8 Barbu Albu Cozma Florea Popa David Zaharia Ursu	2	<p>8 reprezintă numărul concurenților, apoi este memorat numele fiecărui concurent La înscriere avem</p> <table> <tr> <td>Albu</td><td>Barbu</td><td>Cozma</td><td>David</td><td>Florea</td><td>Popa</td><td>Ursu</td><td>Zaharia</td></tr> <tr> <td>2</td><td>1</td><td>3</td><td>6</td><td>4</td><td>5</td><td>8</td><td>7</td></tr> </table>	Albu	Barbu	Cozma	David	Florea	Popa	Ursu	Zaharia	2	1	3	6	4	5	8	7				
Albu	Barbu	Cozma	David	Florea	Popa	Ursu	Zaharia															
2	1	3	6	4	5	8	7															
		<p>După ordonare</p> <table> <tr> <td>Barbu</td><td>Albu</td><td>Cozma</td><td>Florea</td><td>Popa</td><td>David</td><td>Zaharia</td><td>Ursu</td></tr> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p>Lungimea maximă a subșirului comun este 5 Sunt 3 șiruri comune de lungime maximă: 2,3,4,5,8; 2,3,4,5,7; 1,3,4,5,8; 1,3,4,5,7. Va fi ales șirul 2, 3, 4, 5, 8, pentru că este primul subșir crescător comun. Valoarea premiului va fi egală cu 2 deoarece <math>2+3+4+5+8=22</math>, iar <math>22\%5=2</math></p>	Barbu	Albu	Cozma	Florea	Popa	David	Zaharia	Ursu	1	2	3	4	5	6	7	8				
Barbu	Albu	Cozma	Florea	Popa	David	Zaharia	Ursu															
1	2	3	4	5	6	7	8															
5 Lupu Manea Horga Nanu Corbu	3	<p>5 reprezintă numărul concurenților, apoi este memorat numele fiecărui concurent La înscriere avem</p> <table> <tr> <td>Lupu</td><td>Manea</td><td>Horga</td><td>Nanu</td><td>Corbu</td></tr> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table> <p>După ordonare</p> <table> <tr> <td>Corbu</td><td>Horga</td><td>Lupu</td><td>Manea</td><td>Nanu</td></tr> <tr> <td>5</td><td>3</td><td>1</td><td>2</td><td>4</td></tr> </table> <p>Lungimea maximă a subșirului comun este 3 Este un singur subșir comun de lungime maximă: 1,2,4 Valoarea premiului va fi egală cu 3</p>	Lupu	Manea	Horga	Nanu	Corbu	1	2	3	4	5	Corbu	Horga	Lupu	Manea	Nanu	5	3	1	2	4
Lupu	Manea	Horga	Nanu	Corbu																		
1	2	3	4	5																		
Corbu	Horga	Lupu	Manea	Nanu																		
5	3	1	2	4																		

- lungimea fiecărui șir de caractere nu depășește 15 caractere
- $n$  număr natural,  $1 \leq n \leq 100$
- se poate obține și valoarea 0
- Pentru 10% dintre teste se obține un singur subșir comun de lungime maximă  $\leq 3$
- Pentru 10% dintre teste se obțin mai multe subșiruri comune de lungime maximă  $\leq 3$
- Pentru 30% dintre teste se obține un singur subșir comun de lungime maximă  $> 3$
- Pentru 50% dintre teste se obțin mai multe subșiruri comune de lungime maximă  $> 3$

Timp maxim de execuție/test: 1 secundă



**E**xamenul de ATESTAT se adresează absolvenților claselor de matematică-informatică și matematică-informatică, intensiv informatică, *certifică noțiunile teoretice și practice abordate la clasă pe parcursul celor 4 ani de studiu la liceu.*

Pentru aceasta au fost propuse de profesori subiecte astfel :

- **Tehnologia informației și comunicației :** Dimitrievici Lucian Traian, Lăzărescu Andreea Monica și Victoriu Angela
- **Programare :** Popescu Madalina, Joc Genia și Tudor Veronica
- **Baze de date :** Neagu Violeta și Voinea Mirela

## Ce este atestatul la informatica?

**A**testatul este un act prin care se confirmă pregătirea de specialitate/profesională dobândită de absolvenții de liceu, cursuri de zi. El se eliberează absolvenților claselor de matematică-informatică și matematică-informatică intensiv informatică, care au promovat proba de specialitate conform normelor metodologice în vigoare. Atestatul se obține după susținerea și promovarea de către candidați a unei probe de specialitate.

Competențele certificate în urma promovării examenului de atestare profesională sunt:

**a. Pentru absolvenții claselor de matematică informatică, intensiv informatică:**

- I. Realizarea design-ului și structurii produselor soft necesare implementării de: sisteme software, aplicații software, baze de date, pagini WEB. (software orientat client)
- II. Particularizarea, configurarea și modificarea aplicațiilor software, în scopul adaptării sistemelor informaționale ale clientului.

**b. Pentru absolvenții claselor de matematică informatică:**

- I. Realizarea managementului site-urilor WEB, sistemelor de operare a calculatoarelor
- II. Furnizarea facilităților de procesare a datelor, inclusiv a serviciilor suport

Proba de specialitate constă în:

- ↳ **probă practică:** realizarea practică pe calculator a cerințelor din biletul extras în ziua probei. Biletul va cuprinde trei subiecte:

- I. Sistemele de Gestiune a Bazelor de Date
- II. Programare
- III. Sisteme de Operare, abilități de tehnoredactare, calcul tabelar sau prezentări PowerPoint.

- ↳ **proiect:** prezentarea și motivarea teoretică a unui proiect (produs soft) realizat în timpul orelor de laborator, al ultimului an de studiu; produsele pot fi realizate și în echipă (2 - 3 elevi) în funcție de complexitatea proiectului.

Coordonarea examenului de atestare a competențelor profesionale pentru absolvenții se face de către profesorul de informatică de la clasă. În cazul promovării examenului de atestare a competențelor profesionale candidatul primește un atestat. Atestatul se eliberează de secretariatul liceului, în care a fost susținut examenul, după promovarea de către candidat a clasei a XII-a. Eliberarea atestatului nu este condiționată de promovarea examenului de **Bacalaureat**.

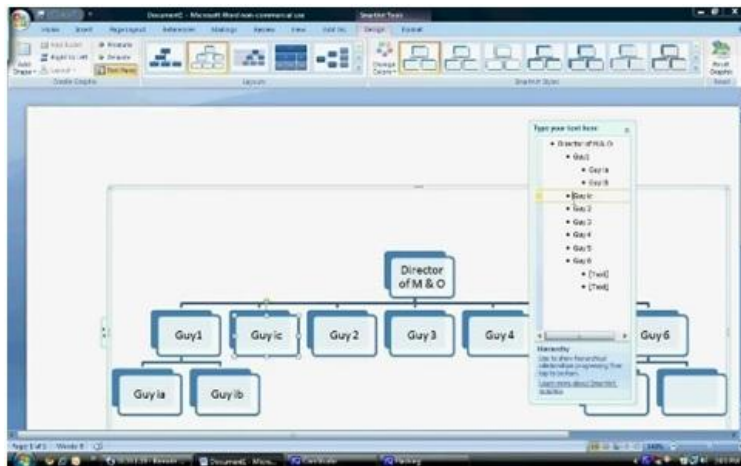




# SUBIECTE OPERARE

## Subiectul nr.1

Folosind aplicația Microsoft Word, în versiunea de pe calculatorul dumneavoastră, realizați un document nou care să descrie componența unei firme mici, cu câțiva angajați, respectiv un manager, un contabil, o secretară și restul muncitori. Realizați organigrama firmei folosind orice tip de diagramă, ca în figura de mai jos:



Salvați documentul cu numele **firma.docx** în directorul **Atestat\_2018** de pe desktop.

## Subiectul nr.2

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E	F	G	H
1	Nr.crt	Nume	Prenume	Localitate	Județ	Vârsta	Produs solicitat	Costuri
2	1	Popescu	Romeo	Bacău	Bacău	10	Skateboard	182
3	2	Ionescu	Sorin	Pitești	Argeș	8	Bicicletă	320
4	3	Georgescu	Ștefan	Curtea de Argeș	Argeș	12	Skateboard	239
5	4	Vasilescu	Vlad	Arad	Arad	18	Pix Parker	3500
6	5	Andreescu	Valentina	Brăila	Brăila	16	Inel	850
7	6	Marinescu	Simona	Dorohoi	Botoșani	15	Brățară argint	250

Sortați datele din tabel în ordine alfabetică după județ, respectiv după localitate.

În celula H8, calculați prețul mediu.

## Subiectul nr.3

Realizați o prezentare pe desktop, **examen\_national.pptx**, formată din două diapozitive, care să conțină următoarele elemente:

Slide-ul 1, de tip titlu, va conține titlul Bacalaureat și subtitlul *Anul școlar 2017/2018*, scrise cu font Tahoma, culoare albastră, dimensiune 30, respectiv 28.

Slide-ul 2, de tip titlu și conținut, va conține în zona de titlu *Probele examenului de bacalaureat*, iar în zona de conținut sunt enumerate printr-o listă cu marcatori, probele examenului național(A,B,C,D,E).



#### Subiectul nr.4

Realizați în Word, un afiș cu numele **Balul\_bobocilor.docx**. Salvați-l pe desktop, în folderul **Atestat\_2018**. Stabiliți formatul de pagină A5, orientarea de tip portret și toate marginile de 1 cm. Acesta va conține: - în antet textul “Avem plăcerea să vă invităm la ”, în document titlul- Balul bobocilor, organizat de....., în locația..... centrat, Times NewRoman, 12 . În subsol inserați data la care se va organiza balul.

#### Subiectul nr.5

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E	F
1	Nr.crt	Nume	Prenume	Clasa	Nr. absențe nemotivate	Măsura adoptată
2	1	Popescu	Adrian	9	23	
3	2	Dontu	Sorin	12	43	
4	3	Georgescu	Ștefan	12	25	
5	4	Colea	Vlad	11	15	
6	5	Andreescu	Valentina	10	5	
7	6	Marin	Simona	9	2	

Pentru completarea coloanei **Clasa**, creați o listă din care să puteți alege valorile 9, 10,11 și 12.

În coloana **Măsura adoptată** introduceți o formulă de calcul care încarcă coloana cu mesaje astfel: **”Mustrare scrisă”** dacă numărul absențelor nemotivate este mai mare sau egal cu 20 dar mai mic ca 40 și elevul este în clasa a 9-a sau a 10-a;”**Avertisment de exmatriculare”** pentru elevii din clasele 11-12 dacă numărul absențelor nemotivate este mai mare sau egal cu 20 dar mai mic ca 40 și **”Exmatriculare”** cazul în care este mai mare sau egal cu 40.Dacă numărul absențelor nemotivate este mai mic de 20, atunci se va afișa mesajul ”OK”.

#### Subiectul nr.6

Creați un fișier numit **Memorie\_externa.pptx**, salvat în folderul **atestat\_info** de pe desktop, având următoarele caracteristici:

- ☞ Slide-ul 1, de tip titlu și conținut, va conține titlul *Cardul de memorie*, iar la conținut, va fi precizată o definiție a acestui dispozitiv de memorare; textul diapozitivului va fi scris cu Times New Roman, dimensiunea 30, respectiv 28.
- ☞ Slide-ul 2, de tip titlu , va conține un link către un site în care sunt menționate avantajele cardurilor de memorie.

#### Subiectul nr.7

Realizați în Word, un document cu numele **formula.docx**. Salvați-l pe desktop, în folderul **Atestat\_2018**. Acesta va conține: - un titlu scris artistic (Binomul lui Newton), fomula următoare, scrisă folosind modulul ecuație



### Subiectul nr.8

Realizați tabelul din figura următoare pentru **Bacalaureat 2018** și salvați fișierul în directorul **Atestat\_2018**, de pe desktop

	A	B	C	D	E	F	G	H
1	Nr.crt	Nume	Prenume	Română	Matematică	Proba 3	Media	Rezultat
2	1	Bobeică	Eugenia	9.25	6.45	8.80		
3	2	Scînteianu	Claudiu	5.70	5.45	6.00		
4	3	Drăgoi	Victor	4.20	10.00	9.45		
5	4	Damian	Elena	6.00	5.80	7.50		
6	5	Orzan	Nicoleta	7.21	3.90	8.00		
7	6	Panait	Cezar	8.30	6.30	7.90		

- ☞ În coloana **Media**, scrieți formula corespunzătoare pentru calculul mediei aritmetice, în cazul în care notele la toate cele trei probe sunt minim 5. În caz contrar, în celula corespunzătoare mediei se va afișa o succesiune de trei liniuțe (---).
- ☞ În coloana **Rezultat** introduceți o formulă de calcul care încarcă coloana cu mesaje astfel: **"Admis"** dacă media este mai mare sau egală cu 6 iar notele la fiecare probă sunt minim 5 și **"Respins"** în caz contrar.

### Subiectul nr.9

În folder-ul **Exercitii** de pe desktop realizați o prezentare numită **Temperamente.pptx**, care cuprinde 2 diapositive, unul în care se menționează definiția temperamentului, iar al doilea să enumere tipurile de temperamente sub forma unui *SmartArt* de tip *Ciclu radial*.

### Subiectul nr.10

Realizați în Word, un document cu numele **Invitatie.docx**, format A4, orientare de tip vedere și toate marginile de 0,5 cm. Salvați-l pe desktop, în folderul **Atestat\_2018**. Trebuie să conțină: - textul „Cursul festiv al clasei a XII-a” scris semicircular și colorat, o imagine adecvată - în partea de jos inserați o formă în care notați: „Cu drag, .....” urmând ca pe puncte să fie scris manual numele colectivului clasei iar în subsol anul curent.

### Subiectul nr.11

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E
1	Nr. crt	Nume	Prenume	Vârsta	Sex
2	1	Măndresc	Liviu	23	M
3	2	Goga	Irina	19	F
4	3	Alexandre	Iris	18	F
5	4	Popa	Monica	20	F
6	5	Grigoresu	Cosmin	20	M
7	6	Grigoriu	Irinel	17	M

- ☞ Pentru completarea coloanei **Sex**, creați o listă din care să puteți alege valorile F sau M.
- ☞ Filtrați tabelul, astfel încât să afișați numai datele persoanelor al căror prenume conține grupul de litere *Iri*.



### **Subiectul nr.12**

Realizați un folder pe desktop cu numele **aplicatie**. În acest folder creați un fișier **date.txt** în care, pe primul rând, se scrie numele elevului, iar pe rândul următor se scrie numărul de fișiere cu extensia **.docx**, găsite pe stația de lucru. Fereastrei cu rezultatele căutării acestor fișiere i se va aplica o captură de ecran salvată în fișierul **dovezi.docx**, construit tot în folder-ul inițial.

### **Subiectul nr.13**

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E
1	Nr. crt	Nume	Prenume	Vârsta	Sex
2	1	Măndresc	Liviu	23	M
3	2	Goga	Irina	19	F
4	3	Alexandre	Iris	18	F
5	4	Popa	Monica	20	F
6	5	Grigoresu	Cosmin	20	M
7	6	Grigoriu	Irinel	17	M

- ☞ Pentru completarea coloanei vârstă să nu permită decât introducerea valorilor situate între 15 și 30. În caz contrar, să se afișeze mesajul de eroare: "Vârsta trebuie să fie între 15 și 30 ani!"
- ☞ Filtrați tabelul, astfel încât să afișați numai datele persoanelor de sex masculin cu vârste de peste 20 ani.

### **Subiectul nr.14**

Realizați în Word, un pliant de prezentare a unui concurs de grafică în format A5, tip portret, cu numele **Poster.docx**. Salvați-l pe desktop, în folderul **Atestat\_2018**. Acesta va conține: pe pagina 1 – numele concursului și tematica acestuia, inserând și o imagine ilustrativă pentru tematica concursului, pe pagina 2 – perioada concursului, cerințele și locația desfășurării concursului, respectiv organizatorii concursului, membrii de jurizare.

### **Subiectul nr.15**

Verificați cerințele calculatorului vostru pentru instalarea unei aplicații software pe sistemul de operare, care necesită minim 1 GB Ram, 750MB de memorie pe hard disk și o frecvență a procesorului de minim 2.0 Ghz. Realizați o captură a resurselor calculatorului pe care să o salvați cu numele **performanțe** și cu orice extensie din tipul **.bmp**, **.jpg**, **.png**.

### **Subiectul nr.16**

Creați pe desktop un director cu numele **Atestat\_2018**. Setați o imagine din calculator ca background pentru desktop, astfel încât să ocupe doar centrul ecranului. Faceți o captură de ecran care să evidențieze setările cerute și salvați-o în directorul **Atestat\_2018**.



### Subiectul nr.17

Creați pe desktop un folder cu numele **Atestat\_2018**. În acesta veți crea următoarele:

- ↵ fișierul **DATELE MELE**, de tip text în care veți scrie numele și prenumele vostru, clasa din care faceți parte și profilul acesteia.
- ↵ arhiva **Rezolvări** care va conține toate fișierele în care ați rezolvat cerințele pentru proba practică a examenului de atestare a competențelor profesionale.

### Subiectul nr.18

Creați un document în care să memorați un paragraf din romanul *Cel mai iubit dintre pământeni*, a lui Marin Preda. Modificați fontul în Times New Roman, mărimea scrisului la 14, scrisul fiind aliniat Justify. Exportați în formatul \*.pdf documentul, cu numele **paragraph. pdf** în directorul **Atestat\_2018**.

### Subiectul nr.19

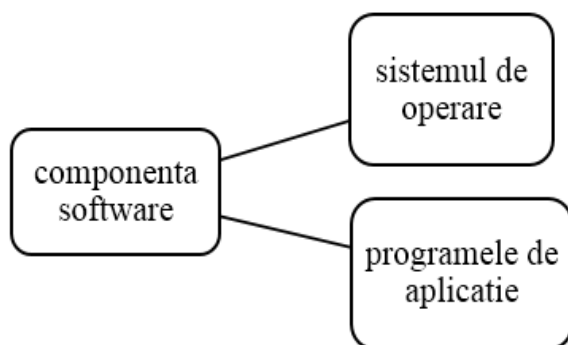
Realizați o prezentare numită **Picturi\_celebre.pptx**, care să conțină următoarele elemente:

- ↵ Primul slide, va avea titlul *Picturi celebre*, scris artistic, iar ca text, va cuprinde informații generale despre picturi celebre.
- ↵ Al doilea slide, va prezenta cele două picturi ale lui Leonardo da Vinci, *Mona Lisa* și *Cina cea de taină*, atât cu imagini sugestive cât și cu descrierea textuală a acestora; textul este scris cu culoarea roșie, respectiv albastră. Fundalul ambelor slide-uri e de tip texturat la alegere.

### Subiectul nr.20

Realizați o prezentare care să conțină două slide-uri cu următoarele elemente:

- ↵ Slide-ul 1: în zona de titlu va fi textul *Componenta software*, iar în subtitlu vor fi menționate numele și prenumele elevului;
- ↵ Slide-ul 2: va conține următoarea diagramă



### Subiectul nr.21

Realizați în Word, un document cu numele **Invoice.docx** - format A4, orientare de tip portret, care să conțină o cerere tip pentru învoirea unui elev într-o zi de școală, precum și motivele învoirii, altele decât cele medicale. Respectați următoarele cerințe: - titlul trebuie să fie centrat, scrise cu fontul Arial Black de mărime 16, îngroșate, scris italic, iar textul din celelalte paragrafe să aibă indentări: stânga 1,2 cm, dreapta 1 cm și spațiere înainte cu 10 puncte. Salvați-l pe desktop, în folderul **Atestat\_2018**.

### Subiectul nr.22

Realizați în Word, un poster de tip Premiera la o piesa de teatru din cadrul unei sesiuni teatrale, în format A5, tip portret, cu numele **Prezentare.docx**. Salvați-l pe desktop, în folderul **Atestat\_2018**. Acesta va avea ca background o imagine sau o textură, numele piesei de teatru va fi scris pe prima linie folosind un scris artistic, pe linia a doua actorii care joacă în piesă, iar pe linia a treia data în care va avea loc premiera.

### Subiectul nr.23




Creați fișierul **atestat\_profesional\_2018.docx** în directorul cu numele vostru de pe desktop, care să cuprindă denumirea lucrării de atestat profesional precum și limbajul de programare folosit, eventual prezentarea. Aliniați titlul la centru și spațiați caracterele acestuia la 5 puncte; enumerați capitolele pe care le conține atestatul într-un cuprins sub acest titlu. Folosiți fontul Times New Roman atât pentru titlu cât și pentru cuprins.

### Subiectul nr.24

Realizați tabelul din figura următoare, care reprezintă încasările a șase magazine pe perioadă de trei luni și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E	F	G
1		Ianuarie		Februarie		Martie	
2		Încasări	Trend	Încasări	Trend	Încasări	Trend
3	Magazin 1	20		30		25	
4	Magazin 2	10		16		30	
5	Magazin 3	50		43		37	
6	Magazin 4	70		25		89	
7	Magazin 5	65		35		45	
8	Magazin 6	15		90		20	

- ↳ Calculați pentru fiecare lună trendurile cu formula  $\text{Încasări}_{(\text{luna curentă})} - \text{Încasări}_{(\text{luna anterioară})}$ , mai puțin pentru februarie.
- ↳ Evidențiați tendințele de creștere, staționare și scădere folosind de la Icon set, săgețile colorate, pentru trenduri astfel:

Icon		Value	Type
	when value is	$\geq$ 50	Number
	when < 50 and	$\geq$ 0	Number
	when < 0		



### Subiectul nr.25

Creați o prezentare care să conțină două slide-uri , respectând cerințele:

Primul slide va conține informațiile din imaginea de mai jos, al doilea slide conține o diagramă bazată pe informațiile din tabelul anterior, diagramă de tip Coloană, având titlul *Statistică stocuri*, iar etichetele de date să fie afișate.

Denumire echipament	Stoc initial	Cantitate vanduta	Stoc final
HDD	15	4	11
Carduri de memorie	23	3	20
Memorie flash	34	23	11

### Subiectul nr.26

Creați un fișier cu numele **Vacanta\_de\_vara.pptx**, salvându-l în subfolder-ul **examen\_12** aflat în folder-ul **aplicatii** de pe desktop. Prezentarea va conține trei slide-uri cu imagini din locații turistice pe fundal, iar în ultimul slide să existe și un link spre o destinație exotica de vacanță. Tranziția slide-urilor se realizează și automat la 3 secunde.

### Subiectul nr.27

Creați un folder pe desktop numit **Competente**, în cadrul căruia se construiesc alte trei sub-foldere, *Competente\_Limba\_romana*, *Competente\_Limba\_engleza*, *Competente\_digitale*. În sub-folderul *Competente\_digitale*, realizați un shortcut spre Microsoft Excel și un fișier în care să se salveze printr-o captură de ecran informații despre modelul și frecvența procesorului, capacitatea memoriei RAM.

### Subiectul nr.28

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E	F	G
1	Nr. Crt	Nume	Prenume	Data nașterii	Vârsta	Data angajării	Vechime
2	1	Grecu	Constantin	3/12/1980		25/4/1998	
3	2	Cristescu	Daniel	11/11/1974		11/11/1994	
4	3	Năstase	Camelia	5/7/1963		5/7/1983	
5	4	Voinea	Ionela	11/9/1966		11/9/1986	
6	5	Ion	Mioara	29/6/1958		29/6/1978	
7	6	Pohrib	Cosmina	30/6/1973		30/6/1990	

- ↳ Completați coloana **Vârsta**, scriind formula corespunzătoare de calcul.
- ↳ Calculați **Vechimea** în ani pentru fiecare angajat și formați astfel încât vechimea mai mare sau egală cu 30 ani să se afișeze automat cu bold, culoare: roșu.

### Subiectul nr.29

Creați fișierul **Bacalaurat\_2018.docx** în directorul cu numele vostru, de pe desktop. Fișierul va conține probele de examen pe care le veti susține sau sunt echivalate prin alte certificări de tip ECDL, Cambridge, etc precum și tipul de examinare (scris-oral) și data susținerii lor, pe câte un rând. Veți folosi o lista neordonată cu un simbol la alegere.





### Subiectul nr. 30

Realizați tabelul din figura următoare, pentru calculul rădăcinilor reale ale ecuației de gradul II și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E	F
1	a	b	c	$\Delta$	x1	x2
2	1	0	4			
3	1	2	1			
4	1	-2	1			
5	4	3	1			
6	-5	6	-8			

- ↪ Calculați discriminantul ecuației de gradul II.
- ↪ Calculați *rădăcinile x1 și x2*, dacă sunt reale. În caz contrar afișat mesajul: **"Ecuația nu are rădăcini reale"**.

### Subiectul nr.31

Realizați un tabel în editorul Microsoft Word care să conțină orarul fictiv al unui elev dintr-o clasă, a unei instituții de învățământ, pentru cele 5 zile de școală. Tabelul va fi cu chenar dublu, datele vor fi centrate și scrise italic cu fontul Verdana, dimensiunea 14. Documentul va conține nota de antet, cu numele instituției de învățământ și de subsol cu numele elevului și profilul de învățământ.

### Subiectul nr.32

Creați o prezentare numită **COMPETENȚE DIGITALE.pptx**, pe desktop. În primul slide, cu tema Organic și aspectul Titlu, scrieți titlul *PROBA B* (font Arial, înclinat, 48, mov), subtitlul- *Bacalaureat 2018*, font Arial, înclinat, 36, albastru, umbrit. În al doilea slide să se insereze o imagine cu profesia la care elevul aspiră. În ambele slide-uri, în subsol, se vor preciza numărul de pagină la dreapta, data și ora sistemului la stânga și numele și prenumele elevului la centru.

### Subiectul nr.33

Creați pe desktop un director cu numele **Atestat\_2018**. Modificați atributele unui fișier astfel încât acesta să fie accesibil doar pentru citire. Folosind instrumentul Snipping Tool, realizați o captură a ferestrei. Proprietăți și salvați-o cu numele **Atribute.png**, în directorul **Atestat\_2018**.

### Subiectul nr.34

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E
1	Nr.crt	Denumire produs	Data expirării	Stare produs	7 octombrie 2018
2	1	Ceapă	5 mai 2018		
3	2	Brânză	23 august 2018		
4	3	Iaurt	12 octombrie 2018		
5	4	Lapte	24 decembrie 2018		
6	5	Chefir	6 mai 2018		
7	6	Salam	7 august 2018		

- ↪ Scrieți o formulă pe baza căreia să completați automat coloana **Stare produs** cu mesajul **"Produs expirat"** dacă data expirării este mai mică decât 07.10.2018 sau **"Produs în termen"** dacă data expirării este mai mare decât 07.10.2018.
- ↪ Sortați alfabetic tabelul după **Denumire produs**.



### Subiectul nr.35

În folder-ul **atestat2018** salvat pe desktop, să se realizeze un fișier numit **Piramide\_egiptene.pptx**, cu trei slide-uri, în care să se prezinte trei piramide, însoțite de imagini semnificative, iar fiecare imagine să reprezinte o legătură către un site ce prezintă informații suplimentare despre piramida precizată. Imaginile vor fi animate cu efect de tip *zbor*.

### Subiectul nr.36

Căutați pe contul curent câteva imagini, maxim 10, de tip ClipArt sau poză și inserați-le într-un document Word, scriindu-le extensiile în dreapta lor, într-un tabel în care pe prima coloană să fie imaginile iar în a doua, extensiile. Salvați documentul Word în format html, în directorul **atestat\_2018** și vizualizați-l ulterior cu un browser web, la alegere.

### Subiectul nr.37

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.

	A	B	C	D	E	F	G
1	Marca	Nume	Prenume	Data nasterii	Sectie	Vechime	Salariu
2	1001	Dumitru	Maria	3/1/1957	3	25	3000
3	1200	Marin	Alina	10/4/1969	2	20	2900
4	1205	Grigorescu	Dan	3/5/1970	1	12	2100
5	1235	Mihala	Dorin	3/20/1978	3	21	2500
6	1302	Stanciu	Alexandru	2/9/1975	3	24	2000
7	1320	Iordache	Livia	6/5/1980	3	2	1500

- ↗ În aceeași foaie de calcul, începând cu celula B20 să se afișeze informațiile referitoare la angajații cu vechime mai mare de 20 de ani care lucrează în secția 3.
- ↗ Trasați un grafic care să prezinte comparativ salariile persoanelor din tabelul anterior. Adăugați graficului următoarele elemente: titlul: **Salarii**, etichete de axe: **Nume** și **Salariu**, etichete de date.

### Subiectul nr.38

Realizați o prezentare formată din două diapozitive cu următoarele elemente:

- ↗ Primul slide va conține ca titlu *Destinații turistice în România*, iar în subtitlu – *Oferta 2018*; textul va fi scris cu culoarea verde, fontul Comic Sans Ms, dimensiunea 40, respectiv 30, fundal texturat la alegere.
- ↗ Al doilea slide va conține informații din tabelul de mai jos

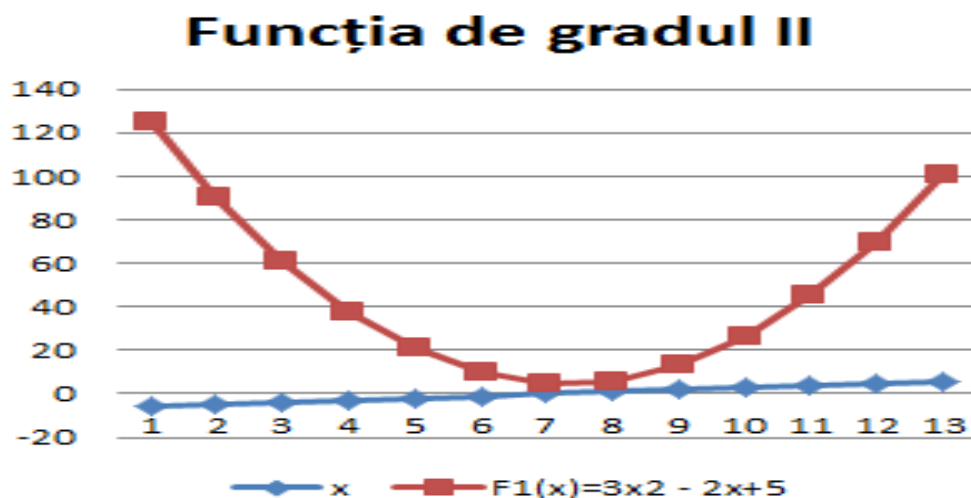
Destinații	Preț
Sinaia	200
Vatra Dornei	300
Cheile Bicazului	350

- ↗ Folosind datele din tabel, realizați o digramă de tip *linie*; modificați culoarea de fundal a graficului în roz.



### Subiectul nr.39

Realizați tabelul din figura următoare și salvați fișierul în directorul **Atestat\_2018**, de pe desktop.



Completați coloana  $F_1(x)$ , scriind formula corespunzătoare de calcul.

Trasați graficul funcției de gradul II, prin puncte cu culori diferite. Adăugați graficului următoarele elemente: titlul, legendă.

### Subiectul nr.40.

Căutați fișierele sursă *c* sau *cpp* de pe contul curent și copiați-le în directorul creat cu numele vostru pe desktop. Creați o arhivă, cu aceste fișiere, folosind un arhivator la alegere, din cele cunoscute. Arhiva va fi parolată folosind parola *atestat*. În final, afișați dimensiunea arhivei, numele ei și parola într-un fișier text în același director.



**Probleme Atestat Informatică 2017-2018 \*Subiecte BAZE DE DATE\* ISJ Galați****Tabela Admitere:**

**Admitere** (nume varchar2(30), nota\_r number(5,2), nota\_m number(5,2), medie\_gim number(5,2), media number(5,2), promovat varchar2(4))

create table Admitere (nume varchar2(30), nota\_r N(5,2), nota\_m N(5,2), medie\_gim N(5,2), media number(5,2), promovat varchar2(4))

Următoarele două probleme fac referire la tabela **Admitere**:

**Subiectul nr 1**

Încărcați tabela cu minim 6 înregistrări, pentru care completați doar numele, nota\_r, nota\_m, medie\_gim.

Să se calculeze și să se completeze media de admitere

$(25\% * \text{medie\_gim} + 75\% * (\text{nota\_r} + \text{nota\_m}) / 2)$

Afișați numărul de elevi promovați.

UPDATE admitere SET media=(nota\_r + nota\_m)/2 \* 0.75 + medie\_gim \*0.25

SELECT count(\*) FROM admitere WHERE UPPER(promovat)='DA'

**Subiectul nr 2**

Încărcați tabela cu minim 6 înregistrări, pentru care completați doar numele, nota\_r, nota\_m, medie\_gim

Să se calculeze și să se completeze media de admitere

$(25\% * \text{medie\_gim} + 75\% * (\text{nota\_r} + \text{nota\_m}) / 2)$

Să se completeze câmpul promovat cu DA, dacă toate mediile sunt  $\geq 5$  și cu NU, dacă măcar o medie este  $< 5$ . Pentru elevii promovați, calculați media generală a notelor de la limba română.

UPDATE admitere SET media=(nota\_r + nota\_m)/2 \* 0.75 + medie\_gim \*0.25

UPDATE admitere SET promovat='DA' WHERE nota\_r $\geq$ 5 AND nota\_m $\geq$ 5 AND medie\_gim $\geq$ 5

UPDATE admitere SET promovat='NU' WHERE nota\_r $<$ 5 OR nota\_m $<$ 5 OR medie\_gim $<$ 5

**Tabela Masini:**

**Masini** (id number(3), denumire varchar2(30), pret number(7), data\_fab date, culoare varchar2(20))

create table Masini (id number(3), denumire varchar2(30), pret number(7), data\_fab date, culoare varchar2(20))

Următoarele două probleme fac referire la tabela **Masini**:



### **Subiectul nr 3**

Afișați mașinile de culoare roșie fabricate înainte de anul 2012.

Afișați mașinile cu prețul mai mare decât prețul mediu.

```
SELECT * FROM masini
WHERE EXTRACT (YEAR FROM data_fab)<2012 AND LOWER(culoare) LIKE 'ros%'
SELECT * FROM masini
WHERE pret >(SELECT AVG(pret) FROM masini)
```

### **Subiectul nr 4**

Să se mărească cu 5% prețul mașinilor fabricate după anul 2012.

Afișați mașinile grupate pe culori, iar pentru mașinile de aceeași culoare, afișarea se va face descrescător după preț.

```
UPDATE masini SET pret= pret+(pret*0.05) WHERE EXTRACT (YEAR FROM data_fab)
>= 2012
```

### **Baza de date Presa**

Conține următoarele tabele:

**Reviste** (*idr* number(3), *titlu* varchar2(30), *pret* number(5,2), *categorie* varchar2(30))

**Abonati** (*ida* number(4), *nume* varchar2(30), *adresa* varchar2(50))

**Abonamente** (*idr* number(3), *ida* number(4), *data\_ab* date)

```
create table Reviste (idr number(3), titlu varchar2(30), pret number(5,2), categorie varchar2(30))
```

```
create table Abonati (ida number(4), nume varchar2(30), adresa varchar2(50))
```

```
create table Abonamente (idr number(3), ida number(4), data_ab date)
```

Următoarele trei probleme fac referire la baza de date **Presa**

### **Subiectul nr 5**

Să se afișeze revistele pe categorii, în ordinea alfabetică a titlurilor.

Să se afișeze numele abonaților și revistele la care sunt abonați.

```
SELECT * FROM reviste ORDER BY categorie, titlu
SELECT nume, titlu
FROM reviste r, abonati a, abonamente b WHERE r.idr=b.idr AND a.ida=b.ida
```

### **Subiectul nr 6**

Afișați revistele descrescător după preț.

Să se afișeze revistele și numele abonaților pentru care abonamentul expiră azi (abonamentul e de 30 de zile calendaristice).

```
SELECT titlu, pret FROM reviste
ORDER BY pret DESC
SELECT nume, titlu FROM reviste r, abonati a, abonamente b
WHERE r.idr=b.idr AND a.ida=b.ida AND data_ab+30 BETWEEN SYSDATE AND
SYSDATE+1
```



### Subiectul nr 7

Care sunt revistele cu prețul maxim?

Care este valoarea totală a revistelor la care s-a abonat Ionescu Vasile?

```
SELECT titlu FROM reviste
WHERE pret=(SELECT MAX(pret) FROM reviste)
SELECT SUM(pret) FROM reviste r, abonati a, abonamentee b
WHERE r.idr=b.idr AND a.ida=b.ida AND UPPER(ume)='IONESCU VASILE'
```



### Tabela Angajati:

**Angajati** (id number(3), nume varchar2(30), data\_nas date, data\_ang date, salariu number(5))

create table **Angajati** (id number(3), nume varchar2(30), data\_nas date, data\_ang date, salariu number(5))

Următoarele două probleme fac referire la tabela Angajati:

### Subiectul nr 8

Să se afișeze angajații cu o vechime mai mare de 10 ani.

Să se afișeze numele celui mai vechi angajat.

```
SELECT nume FROM angajati
WHERE (SYSDATE-data_ang)/365>=10
S E L E C T      n u m e      F R O M      a n g a j a t i
WHERE data_ang=(SELECT MIN(data_ang) FROM angajati)
```

### Subiectul nr 9

Să se mărească cu 10% salariile mai mici de 2000 lei.

Să se afișeze angajații în ordinea descrescătoare a salariilor.

```
UPDATE angajati SET salariu=salariu*1.1 WHERE salariu<2000
SELECT nume, salariu FROM angajati ORDER BY salariu DESC
```

### Baza de date Magazin

Conține următoarele tabele:

**Produse** (idp number(3), denumire varchar2(30), pret number(5,2), cantitate number(4), categorie varchar2(30))

**Furnizori** (idf number(4), nume varchar2(30), localitate varchar2(50))

**Comenzi** (idpnumber(3), idf number(4), data\_com date)

create table Produse (idp number(3), denumire varchar2(30), pret number(5,2), cantitate number(4), categorie varchar2(30))

create table Furnizori (idf number(4), nume varchar2(30), localitate varchar2(50))

create table Comenzi (idp number(3), idf number(4), data\_com date)



### Subiectul nr 10

Să se afișeze toți furnizorii din Galați

Din ce localități sunt furnizorii de produse din categoria *sport*?

```
SELECT * FROM furnizori
WHERE UPPER(localitate) LIKE '%GALATI%'
SELECT localitate FROM produse p, furnizori f, comenzi c
WHERE LOWER(categorie)='sport' AND p.idp=c.idp AND f.idf=c.idf
```



### Subiectul nr 11

Să se afișeze numărul de produse care nu mai sunt pe stoc

Să se afișeze furnizorii produselor cu preț maxim

```
SELECT COUNT (*) FROM produse
WHERE cantitate=0
SELECT nume FROM produse p, furnizori f, comenzi c
WHERE p.idp=c.idp AND f.idf=c.idf AND pret=(SELECT MAX(pret) FROM produse)
```

### Subiectul nr 12

Să se afișeze prețul mediu al produselor din categoria *cosmetice*

Să se afișeze valoarea totală a produselor furnizate de firma Napolact.

```
SELECT AVG(pret) FROM produse WHERE LOWER(categorie)='cosmetice'
SELECT SUM(pret*cant) FROM produse p, furnizori f, comenzi c
WHERE p.idp=c.idp AND f.idf=c.idf AND UPPER(nume)='NAPOLACT'
```

### Subiectul nr 13

Să se afișeze valoarea totală a produselor care au prețul cuprins între 20 și 30 lei

Să se afișeze produsele comandate în ultimele 3 zile și furnizorii acestora.

```
SELECT SUM(pret*cantitate) FROM produse WHERE pret>=20 AND pret<=30
SELECT denumire, nume FROM produse p, furnizori f, comenzi c
```

### Baza de date Biblioteca

Conține următoarele tabele:

**carti** (*id* number(3), *titlu* varchar2(40), *autor* varchar2(30), *editura* varchar2(30), *an\_aparitie* number(4), *pret* number(5))

**cititori** (*idc* number(4), *nume* varchar2(10), *data\_n* date, *loc* varchar2(20), *mail* varchar2(30) )

**imprumut** (*id* number(3), *idc* number(4), *data\_impr* date)

```
create table carti (id number(3), titlu varchar2(40), autor varchar2(30), editura varchar2(30),
an_aparitie number(4), pret number(5))
```

```
create table cititori (idc number(4), nume varchar2(10), data_n date, loc varchar2(20), mail varchar2
(30) )
```

```
create table imprumut (id number(3), idc number(4), data_impr date)
```

Următoarele cinci probleme fac referire la baza de date Biblioteca





#### **Subiectul nr 14**

Să se afișeze numele cititorilor care au împrumutat cărți în anul 2018

Să se afișeze cărțile cu prețul mai mare decât prețul mediu

```
SELECT nume FROM cititori c, imprumut i
WHERE c.idc=i.idc AND EXTRACT(YEAR FROM data_impr)=2018
SELECT titlu FROM carti
WHERE pret>(SELECT AVG(pret) FROM carti)
```

#### **Subiectul nr 15**

Să se afișeze numărul de edituri

Să se afișeze numele cititorilor care au împrumutat cea mai scumpă carte

```
SELECT COUNT(DISTINCT editura) FROM carti
SELECT nume FROM carti c, cititori t, imprumut i
WHERE c.idc=i.idc AND t.id=i.id AND pret>(SELECT MAX(pret) FROM carti)
```

#### **Subiectul nr 16**

Să se afișeze valoarea cărților de la editura All

Să se afișeze cititorii al căror nume începe cu litera P și care au împrumutat cărți de Ion Creangă

```
SELECT SUM(pret) FROM carti
WHERE UPPER(editura) = 'ALL'
SELECT nume FROM carti c, cititori t, imprumut i
WHERE c.idc=i.idc AND t.id=i.id AND nume LIKE 'P%' AND autor='Ion Creanga'
```

#### **Subiectul nr 17**

Să se afișeze numele cititorilor care au adresă de mail pe gmail.com

Să se afișeze prețul cărților împrumutate de cititori din Galați

```
SELECT nume FROM cititori WHERE mail LIKE '%@gmail.com'
SELECT SUM(pret) FROM carti c, cititori t, imprumut i
WHERE c.idc=i.idc AND t.id=i.id AND loc='Galati'
```

#### **Subiectul nr 18**

Să se afișeze numele cititorilor, descrescător după data la care au împrumutat cărți

Să se afișeze numele cititorilor care au împrumutat cărți luna aceasta

```
SELECT nume, data_impr FROM cititori c, imprumut i
WHERE c.id=i.id
ORDER BY data_impr DESC
SELECT nume, data_impr FROM cititori c, imprumut i
WHERE c.id=i.id
AND EXTRACT (MONTH FROM data_impr) = EXTRACT (MONTH FROM SYSDATE)
AND EXTRACT(YEAR FROM data_impr)= EXTRACT(YEAR FROM SYSDATE)
```



### **Baza de date Auto**

**Conține următoarele tabele:**

**proprietari** (*cnp* VARCHAR2(13), *nume* VARCHAR2(20), *telefon* VARCHAR2(10), *nr\_inmatr* VARCHAR2(9))

**autovehicule** (*nr\_inmatr* VARCHAR2(9), *marca* VARCHAR2(6), *model* VARCHAR2(10), *culoare* VARCHAR2(10), *an\_fabr* NUMBER(4,0))

create table proprietari (cnp VARCHAR2(13), nume VARCHAR2(20), telefon VARCHAR2(10), nr\_inmatr VARCHAR2(9))

create table autovehicule (nr\_inmatr VARCHAR2(9), marca VARCHAR2(6), model VARCHAR2(10), culoare VARCHAR2(10), an\_fabr NUMBER(4,0))

**Următoarele două probleme fac referire la baza de date Auto**

### **Subiectul nr 19**

Să se afișeze mărcile autovehiculelor (fără a se repeta)

Să se afișeze numerele de înmatriculare ale mașinilor cu o vechime mai mică de 10 ani.

```
SELECT DISTINCT marca FROM autovehicule
SELECT p.nr_inmatr, nume FROM proprietari p, autovehicule a
WHERE p.nr_inmatr = a.nr_inmatr
AND EXTRACT(YEAR FROM SYSDATE)-an_fabr<10
```

### **Subiectul nr 20**

Să se afișeze toți proprietarii de mașini cu număr de înmatriculare de Galați (GL).

Să se afișeze numele proprietarului cu cea mai veche mașină și marca și modelul acesteia.

```
SELECT nume FROM proprietari WHERE nr_inmatr LIKE 'GL%'
SELECT nume, marca, model FROM proprietari p, autovehicule a
WHERE p.nr_inmatr = a.nr_inmatr AND an_fabr=(SELECT MIN(an_fabr) FROM
```

### **Baza de date Cinema**

**Conține următoarele tabele:**

**Filme** (*idf* number(3),*nume* varchar2(40),*gen* varchar2(20), *buget* number(9), *anaparitie* number(4))

**Actori** (*ida* number(3),*nume* varchar2(40),*datanastere* Date)

**Distribuție** (*idf* number(3), *ida* number(3))

create table Filme (*idf* number(3),*nume* varchar2(40),*gen* varchar2(20), *buget* number(9), *anaparitie* number(4) )

create table Actori (*ida* number(3),*nume* varchar2(40),*datanastere* Date)

create table Distribuție (*idf* number(3), *ida* number(3))

**Următoarele șapte probleme fac referire la baza de date Cinema**



### **Subiectul nr 21**

Să se afișeze filmele ordonate după gen iar în cadrul aceluiasi gen vor fi ordonate după buget

Să se afișeze în ordine alfabetică actorii care au jucat in filmul **Ultimii Jedi**

```
SELECT * FROM filme ORDER BY gen, buget
SELECT a.numa
FROM filme f, actori a, distributie d
WHERE f.idf=d.idf AND a.ida=d.ida AND lower(f.numa) LIKE '%ultimii jedi%'
ORDER BY a.numa
```

### **Subiectul nr 22**

Să se afișeze filmele din genul **aventură** ordonate după buget

Să se afișeze în ordine alfabetică actorii care au jucat in filme apărute în anul  
**2017**

```
SELECT * FROM filme WHERE gen='aventura' ORDER BY buget
SELECT a.numa
FROM filme f, actori a, distributie d
WHERE f.idf=d.idf AND a.ida=d.ida AND anaparitie=2017
ORDER BY a.numa
```



### **Subiectul nr 23**

Să se afișeze care a fost cel mai mare buget alocat pentru un film apărut în anul **2017**

Să se afișeze lunile în care s-au născut actorii care au jucat în filme apărute în anul **2010**

```
SELECT MAX(buget) FROM filme WHERE anaparitie=2017
SELECT a.numa, to_char(datanastere,'MONTH')
FROM filme f, actori a, distributie d
WHERE f.idf=d.idf AND a.ida=d.ida AND anaparitie=2010
```

### **Subiectul nr 24**

Să se afișeze câte filme sunt în baza de date

Să se afișeze actorii născuți în 1990 care au jucat în filme

```
SELECT count(*) FROM filme
SELECT a.numa FROM filme f, actori a, distributie d
WHERE f.idf=d.idf AND a.ida=d.ida AND to_char(a.datanastere,'YYYY')=1990
```

### **Subiectul nr 25**

Să se afișeze anul nașterii fiecărui actor din baza de date

Să se determine câți actori au jucat în filmele din 2017

```
SELECT numa, to_char(datanastere,'YYYY') FROM actori
SELECT COUNT(a.numa)
FROM filme f, actori a, distributie d
```



### **Subiectul nr 26**

Să se determine bugetul mediu pentru fiecare categorie de filme

Să se afișeze în ordine alfabetică actorii care au jucat in filme din seria **Star Wars**

```
SELECT gen, AVG(buget) FROM filme GROUP BY gen
SELECT a.numa
FROM filme f, actori a, distributie d
WHERE f.idf=d.idf AND a.ida=d.ida AND lower(f.numa) LIKE '%star wars%'
ORDER BY a.numa
```

### **Subiectul nr 27**

Să se afișeze filmul care a avut cel mai mare buget în 2017

Să se afișeze cel mai tânăr actor care a fost distribuit într-un film în 2017

```
SELECT * FROM filme
WHERE buget=(SELECT max(buget) FROM filme WHERE anaparitie=2017)
SELECT nume FROM actori
WHERE datanastere=(SELECT max(datanastere)
FROM filme f, actori a, distributie d
WHERE f.idf=d.idf AND a.ida=d.ida AND anaparitie=2017)
```

### **Baza de date Spital**

Conține următoarele tabele:

**Pacienti** (*idp* number(3), *numa* varchar2(40), *salon* number(3), *afectiune* varchar2(40), *datainternare* Date)

**Medicamente** (*idm* number(3), *denumire* varchar2(50), *categorie* varchar2(20), *pret* number(8,2))

**Tratamente** (*idp* number(3), *idm* number(3), *ora* number(2))

create table **Pacienti** (**idp** number(3), **numa** varchar2(40), **salon** number(3), **afectiune** varchar2(40),datainternare Date)

create table **Medicamente** (**idm** number(3), **denumire** varchar2(50), **categorie** varchar2(20), **pret** number(8,2))

create table **Tratamente** (*idp* number(3), *idm* number(3), *ora* number(2))

Următoarele șase probleme fac referire la baza de date Spital

### **Subiectul nr 28**

Să se afișeze în ordine alfabetică numele pacienților și afecțiunile lor pentru salonul 1.

Să se determine costul tratamentului pentru o zi pentru pacientul **Avram** de la salonul cu numărul1.

```
SELECT nume, afectiune FROM pacienti WHERE salon=1 ORDER BY nume
SELECT sum(pret) FROM pacienti p, tratamente t, medicamente m
WHERE p.idp=t.idp AND m.idm=t.idm AND lower(p.numa) like 'avram' and salon=1
```



### **Subiectul nr 29**

Să se afișeze în ordine alfabetică o singură dată lista afecțiunilor pacienților din baza de date

Să se afișeze tratamentul: medicația și ora pentru pacientul **Costescu** de la salonul cu numărul **2**

```
SELECT DISTINCT afectiune FROM pacienti ORDER BY afectiune
SELECT m.denumire, t.ora FROM pacienti p, tratamente t, medicamente m
WHERE p.idp=t.idp AND m.idm=t.idm AND lower(p.num) like 'costescu' and salon=2
```

### **Subiectul nr 30**

Să se afișeze prețul mediu pentru fiecare categorie de medicamente

Să se afișeze pacienții, medicamentele lor din categoria *analgezice* și ora tratamentului

```
SELECT categorie, AVG(pret) FROM medicamente GROUP BY categorie
SELECT p.num, m.denumire, t.ora FROM pacienti p, tratamente t, medicamente m
WHERE p.idp=t.idp AND m.idm=t.idm AND categorie='analgezic'
```

### **Subiectul nr 31**

Să se afișeze alfabetic pacienții internați în luna *februarie*

Să se afișeze costurile tratamentelor pentru fiecare salon

```
SELECT* FROM pacienti
WHERE to_char(datainternare,'Month')='February' ORDER BY nume
SELECT salon, sum(pret) FROM pacienti p, tratamente t, medicamente m
WHERE p.idp=t.idp AND m.idm=t.idm GROUP BY salon
```

### **Subiectul nr 32**

Să se afișeze pacienții pe saloane crescător și alfabetic după nume

Să se afișeze costul medicației fiecărui pacient

```
SELECT salon, nume FROM pacienti
ORDER BY salon, nume
SELECT nume, sum(pret) FROM pacienti p, tratamente t, medicamente m
WHERE p.idp=t.idp AND m.idm=t.idm GROUP BY nume
```

### **Subiectul nr 33**

Să se afișeze numele pacienților și saloanele lor precum și denumirea medicamentelor pentru tratamentele de la ora 16. Datele vor fi ordonate pe saloane.

Să se afișeze modifice numele pacientului **Avram Mihai** de la salonul **1** în **Avramescu Mihai**

```
SELECT nume, denumire, salon
FROM pacienti p, tratamente t, medicamente m
WHERE p.idp=t.idp AND m.idm=t.idm AND ora=16 ORDER BY salon
UPDATE pacienti SET nume = 'Avramescu Mihai'
WHERE lower(num)='% avram% mihai%' AND salon=1
```



### **Baza de date Parcare**

**Conține următoarele tabele:**

**clienti** (*idc* number(3), *nume* varchar2(40), *adresa* varchar2(40))

**automobile** (*ida* number(3), *numar\_inmatriculare* varchar2(10), *marca* varchar2(20), *an\_fabricatie* number(4) )

**locuri** (*idc* number(3), *ida* number(3), *numar\_loc* number(3))

```
create table clienti (idc number(3), nume varchar2(40), adresa varchar2(40))
```

```
create table automobile (ida number(3), numar_inmatriculare varchar2(10), marca varchar2(20),  
an_fabricatie number(4) )
```

```
create table locuri (idc number(3), ida number(3), numar_loc number(3))
```

**Următoarele patru probleme fac referire la baza de date Parcare**

#### **Subiectul nr 34**

Să se determine câte mașini sunt în parcare

Să se afișeze numele proprietarilor de mașini înmatriculate în **Galați**, numărul de înmatriculare și locul de parcare

```
SELECT count(*) FROM automobile  
SELECT nume, numar_inmatriculare, numar_loc  
FROM clienti c, automobile a, locuri l  
WHERE l.ida=a.ida AND l.idc=c.idc AND UPPER(substr(numar_inmatriculare, 1,2))  
='GL'
```

#### **Subiectul nr 35**

Să se afișeze o singură dată alfabetic numele proprietarilor de mașini

Să se afișeze numele proprietarilor, locul de parcare, numărul de înmatriculare și marca mașinii.  
Datele vor fi ordonate după numărul locului de parcare.

```
SELECT DISTINCT nume FROM persoane ORDER BY nume  
SELECT nume, numar_loc, numar_inmatriculare, marca  
FROM clienti c, automobile a, locuri l  
WHERE l.ida=a.ida AND l.idc=c.idc ORDER BY numar_loc
```

#### **Subiectul nr 36**

Să se determine câte mașini înmatriculate în Galați sunt în parcare

Să se afișeze numele proprietarului, marca și locul de parcare pentru autoturismul cu numărul de înmatriculare **GL-03-ADI**

```
SELECT COUNT(*) FROM automobile  
WHERE UPPER(substr(numar_inmatriculare, 1,2))='GL'  
SELECT nume, marca, numar_loc  
FROM clienti c, automobile a, locuri l  
WHERE l.ida=a.ida AND l.idc=c.idc AND UPPER(numar_inmatriculare)  
LIKE '%GL%03%ADI'
```





### **Subiectul nr 37**

Să se afișeze o singură dată mărcile mașinilor fabricate în 2017 ordonate alfabetic

Să se afișeze datele tuturor mașinilor deținute de **Duca Mihai**: număr de înmatriculare, marca și loc de parcare.

```
SELECT DISTINCT marca FROM automobile
WHERE an_fabricatie = 2017 ORDER BY marca
SELECT numar_inmatriculare , marca, numar_loc
FROM clienti c, automobile a, locuri l
WHERE l.ida=a.ida AND l.idc=c.idc AND lower(ume) LIKE '%duca% mihai%'
```

### **Subiectul nr 38**

Să se afișeze numărul de mașini din fiecare an de fabricație, ordonate după anul de fabricație

Să se actualizeze cu valoarea 10 numărul locului de parcare pentru automobilul cu numărul de înmatriculare GL-03-ADI.

```
SELECT an_fabricatie, COUNT(*) FROM automobile ORDER BY an_fabricatie, GROUP BY
an_fabricatie
UPDATE locuri SET numar_loc = 10
WHERE numar_loc=
(SELECT numar_loc FROM clienti c, automobile a, locuri l
WHERE l.ida=a.ida AND l.idc=c.idc
AND UPPER(numar_inmatriculare) LIKE '%GL%03%ADI')
```

### **Subiectul nr 39**

Să se afișeze numerele de înmatriculare și mărcile mașinilor ordonate pe județe

Să se elibereze locul de parcare al autoturismului cu numărul de înmatriculare GL-03-ABC

```
SELECT numar_inmatriculare, marca
FROM automobile
ORDER BY numar_inmatriculare
DELETE FROM locuri
WHERE numar_loc=(SELECT numar_loc
FROM clienti c, automobile a, locuri l
WHERE l.ida=a.ida AND l.idc=c.idc
AND UPPER(numar_inmatriculare) LIKE '%GL%03%ABC')
```

### **Subiectul nr 40**

Să se afișeze ordonați după nume clienții care au adresa în **Galați**

Să se afișeze numărul de înmatriculare, marca și numele proprietarului pentru locul de parcare cu numărul 2

```
SELECT * FROM clienti
WHERE lower(adresa) LIKE '%galati%' ORDER BY nume
SELECT numar_inmatriculare , marca, nume
FROM clienti c, automobile a, locuri l
WHERE l.ida=a.ida AND l.idc=c.idc AND numar_loc=2
```



## Subiectul nr 1

În fișierul text ATESTAT.IN se găsesc, pe o singură linie, separate prin câte un spațiu, mai multe numere naturale de cel mult 9 cifre fiecare. Să se scrie în fișierul ATESTAT.OUT separate printr-un spațiu, ultimele două numere pare (nu neapărat distincte) din fișierul ATESTAT.IN. Dacă în fișier se găsește un singur număr par sau niciun număr par se va scrie în fișierul ATESTAT.OUT mesajul Numere insuficiente.

**Exemplu** dacă fișierul ATESTAT.IN conține valorile: 12 15 68 13 17 90 31 42  
fișierul ATESTAT.OUT va conține valorile: 90 42 .

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("ATESTAT.IN");
ofstream g("ATESTAT.OUT");
long par1, par2, x;
int main()
{par1=par2=-1;
while(f>>x)
    if(x%2==0)
    {par1=par2;
    par2=x;}
if (par1==-1)
    g<<"Numere insuficiente";
else
    g<<par1<<" "<<par2;
return 0;}
```

## Subiectul nr 2

Se citește un șir de numere naturale, de cel mult patru cifre fiecare, până la întâlnirea numărului 123. Să se afișeze acele numere care au cifra de control impară.

**Exemplu:** Pentru șirul de numere 12, 33, 128, 345, 123 se vor afișa valorile 12, 128.

```
#include <iostream>
using namespace std;
int main()
{ unsigned int x, aux, c;
cin>>x;
while(x!=123)
{
    aux=x;
    if(aux<=9) c=aux;
    else if(aux%9!=0) c=aux%9;
    else c=9;
    if(c%2) cout<<x<<" ";
    cin>>x;
}
return 0;}
```



### **Subiectul nr 3**

Fişierul text ATESTAT.IN conţine un număr natural  $n$  ( $0 < n \leq 50$ ) şi număr natural  $k$  ( $0 < k \leq 20$ ). Scrieţi programul C/C++ care determină şi scrie în fişierul ATESTAT.OUT, în ordine descrescătoare, primele  $n$  numere naturale nenule divizibile cu  $k$ . Numerele vor fi separate prin câte un spaţiu.

**Exemplu** Dacă fişierul ATESTAT.IN conţine valorile: 3 5 , atunci fişierul ATESTAT.OUT va conţine numerele : 15 10 5

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("ATESTAT.IN");
ofstream g("ATESTAT.OUT");
int n, i, k;
int main()
{f>>n>>k;
for(i=n; i>=1;i--)
    g<<i*k<<" ";
    return 0;
}
```

### **Subiectul nr 4**

Din fişierul TEXT.IN se citeşte un text. Scrieţi un program care afişează pe ecran numărul vocalelor din text şi transformă vocalele din şir în majuscule.

**Exemplu:**

Din fişierul TEXT.IN se citeşte "Maine am teza la informatica"

Se va afişa:12 vocale şi "MAInE Am tEzA lA InfOrmAtIcA"

```
#include <iostream>
#include <string.h>
#include <fstream>
using namespace std;
int main()
{ int i,nr=0;
    char Text[256],voc[]="aeiouAEIOU";
    ifstream f("date.in");
    f.get(Text,256);
    for(i=0;i<strlen(Text);i++)
        if(strchr(voc,Text[i])) { nr++;
        Text[i]=toupper(Text[i]); }
    cout<<"Cuvantul are "<<nr<<" vocale.";
    cout<<endl<<Text;
    return 0;}
```



### **Subiectul nr 5**

Fișierul text ATESTAT.IN conține un număr natural  $n$  ( $n \leq 500$ ) și apoi  $n$  cifre separate prin spații. Scrieți programul C/C++ care determină și scrie în fișierul ATESTAT.OUT cele  $n$  cifre citite, în ordine crescătoare, separate prin câte un spațiu.

#### **Exemplu**

Dacă fișierul ATESTAT.IN conține numărul 19 și cifrele 3 3 0 9 2 1 2 1 3 7 1 5 2 7 1 0 3 2 3 , atunci fișierul ATESTAT.OUT va conține: 0 0 1 1 1 1 2 2 2 2 3 3 3 3 3 5 7 7 9

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("ATESTAT.IN");
ofstream g("ATESTAT.OUT");
int n, i, j, x, fr[12];
int main()
{
    f >> n;
    for(i=1; i<=n; i++)
    {
        f >> x; fr[x]++;
    }
    for(i=0; i<=9; i++)
        for(j=1; j<=fr[i]; j++)
            g << i << " ";
    return 0;
}
```

### **Subiectul nr 6**

Se citește un număr natural  $n$ ,  $n \leq 30$ , apoi un șir de  $n$  numere întregi de cel mult 5 cifre. Să se afișeze intervalul  $[x, y]$ , care conține toți termenii șirului citit.

**Exemplu:** pentru  $n=5$  și șirul de numere -23, 45, -8, -123, 2 se va afișa  $[-123, 2]$ .

```
#include <iostream>
using namespace std;
int main()
{
    long n, i, x, maxi, mini;
    cin >> n; cin >> x; maxi=mini=x;
    for( i=2; i<=n; i++)
    {
        cin >> x;
        if(x>maxi) maxi=x;
        if(x<mini) mini=x;
    }
    cout << '[' << mini << ', ' << maxi << ']' ;
    return 0;
}
```



### Subiectul nr 7

Se citește un număr natural  $n$ ,  $n \leq 100$ . Să se afișeze cel mai apropiat număr de  $n$ , număr care are suma cifrelor un număr perfect.

(Un număr este perfect dacă el este egal cu suma divizorilor săi, fără numărul însuși. Exemplu: 6 este număr perfect pentru că  $6=1+2+3$ ).

**Exemplu:** Pentru  $n=12$  se va afișa 6.

```
#include <iostream>
using namespace std;
int main()
{ unsigned int n,x, ok=0;
  cin>>n;
  x=n-1;
  while( ok==0 && x>0)
  {   long aux=x; long s=0;
      while(aux)
      {   s+=aux%10; aux/=10; }
      long s1=0;
      for( int d=1;d<x;d++)
          if(x%d==0) s1+=d;
      if(s==s1) ok=1;
      else x--; }
  cout<<x;   return 0; }
```

### Subiectul nr 8

Din fișierul DATE.IN se citește un text format din cuvinte separate de câte un spațiu. Să se afișeze în fișierul DATE.OUT cuvintele din text care au cel puțin 3 consoane sau 3 vocale consecutive.

**Exemplu:**

DATE.IN: Creionul strălucitor desenează frumos pe foaia aceea.

DATE.OUT: Creionul strălucitor foaia aceea

```
#include <fstream>
#include <string.h>
using namespace std;
int main()
{ char voc[6]="aeiou" , cuv[200];
  int i,n,ok,k;
  ifstream f ("date.in");
  ofstream g ("date.out");
  while (!f.eof())
  { f.get(cuv,200,' '); f.get();
    n=strlen(cuv);
    k=0; ok=0;
    for(i=0;i<n;i++)
    { if(strchr(voc,cuv[i]) && k<3) k++;
      else k=0;
      if(k==3) {g<<cuv<<" "; ok=1;}
    }
    if(ok==0)
    { for(i=0;i<n;i++)
      { if(strchr(voc,cuv[i])==0 && k<3) k++;
        else k=0;
        if(k==3) {g<<cuv<<" "; ok=1;}
      }
    }
  }
  return 0; }
```

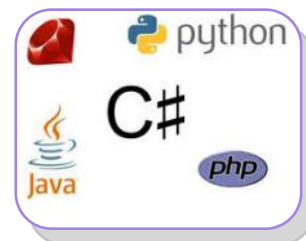


### Subiectul nr 9

Se citește un număr natural  $n$ , de maxim 9 cifre. Să se afișeze cel mai mare număr natural format cu cifrele lui  $n$ .

**Exemplu:** Pentru  $n=233984445$  se va afișa numărul 985444332.

```
#include <iostream>
using namespace std;
int v[10], i;
int main()
{ long n, x, j;
  cin >> n;
  while (n)
  { v[n%10]++; n=n/10; }
  for (i=9; i>=0; i--)
  { x=v[i];
    for (j=1; j<=x; j++) cout << i; }
  return 0; }
```



### Subiectul nr 10

Fișierul text ATESTAT.IN conține un număr natural  $n$  ( $1 < n < 100$ ) și  $n$  componente numere naturale ale unui vector. Scrieți un program care determină și afișează componentele cu cei mai mulți divizori. Se va folosi un subprogram pentru citirea vectorului și unul pentru calculul numărului de divizori ai unui număr.

**Exemplu** Pentru  $n = 5$  și vectorul  $v = \{2, 18, 6, 8, 4\}$ , se va afișa: 18

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("ATESTAT.IN");
int n, i, j, maxi, v[100];
void citire()
{ f >> n;
  for (i=1; i<=n; i++)
    f >> v[i];
}
int divizori(int x)
{ int nr=0;
  for (int j=1; j<=x; j++)
    if (x%j==0)
      nr++;
  return nr;
}
int main()
{ citire();
  maxi=divizori(v[1]);
  for (i=2; i<=n; i++)
    if (maxi<divizori(v[i]))
      maxi=divizori(v[i]);
  for (i=1; i<=n; i++)
    if (maxi==divizori(v[i]))
      cout << v[i] << " ";
  return 0; }
```





### Subiectul nr 11

Construiți o matrice cu  $n \times n$  componente de tip character. Matricea va fi construită astfel: pe diagonala principală va conține '\*', deasupra diagonalei principale '!', iar sub diagonala principală caracterul '?'. Să se afișeze matricea construită.

**Exemplu:** Pentru  $n=4$  se va afișa:

```
*!!!
?*!!
??*!
???*

#include<iostream>
using namespace std;
int main()
{ int i, j, n;
  char a[100][100];
  cout<<"n="; cin>>n;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      { if(i>j) a[i][j] = '!';
        if(i==j) a[i][j] = '*';
        if(i<j) a[i][j] = '?';
      }
  for(i=1;i<=n;i++)
    {for(j=1;j<=n;j++)
      cout << a[i][j];
      cout << endl;
    }
  return 0;}
```



### Subiectul nr 12

Scrieți un program care determină suma elementelor pare dintr-un vector  $x$  cu  $n$  numere naturale. Se citesc de la tastatură  $n$  și elementele vectorului. Se afișează suma sau un mesaj corespunzător dacă nu există elemente pare.

**Exemplu** Pentru  $n = 5$  și  $x = \{12, 9, 45, 3, 22\}$ , se afișează: 34

```
#include <iostream>
using namespace std;
int n, i, v[100], s, ok;
int main()
{  cin>>n;
   for(i=1; i<=n; i++)
     cin>>v[i];
   s=0;
   for(i=1; i<=n; i++)
     if(v[i]%2==0)
       { s=s+v[i];      ok=1;
       }
   if(ok)
     cout<<s;
   else
     cout<<"Nu exista numere pare";
   return 0;
}
```



### **Subiectul nr 13**

Se consideră un șir  $v$  de  $n$  numere întregi. Scrieți un program care verifică dacă o valoare  $x$ , citită de la tastatură, apare sau nu în vector. În caz afirmativ, se va preciza și poziția în care valoarea apare în șir. În caz contrar, se va afișa mesajul NU.

**Exemplu** Pentru  $n = 5$ ,  $v = \{3, 12, 1, 4, 13\}$  și  $x = 13$  se afișează: Valoarea 13 apare în poziția 5, iar pentru aceleași valori și  $x = 20$ , se afișează: NU

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

### **Subiectul nr 14**

Se citește  $p$  un număr natural. Construiți în memorie o matrice pătratică cu  $n$  linii și  $n$  coloane, unde  $n$  este numărul de cifre al numărului citit. Matricea va fi construită astfel: pe diagonala principală va conține **0**, deasupra diagonalei principale cea mai mare cifră a numărului  $p$ , iar sub diagonala principală cea mai mică cifră a numărului  $p$ . Matricea va fi afișată pe următoarele  $n$  linii elementele fiecărei linii fiind separate de câte un singur spațiu.

**Exemplu:** Dacă se citește pentru  $p=824$  se va afișa:

**0 8 8**

**2 0 8**

**2 2 0**

```
#include<iostream>
using namespace std;
int main()
{
    int i, j, a[100][100], p, mn=0, mx=0, nrCifre=0;
    cout<<"p=";cin >> p; mn = p%10; mx = p%10;
    while(p > 0)
    {
        if(p % 10 > mx) mx = p%10;
        else if(p%10 < mn) mn = p%10;
        nrCifre++;
        p=p/10;
    }
    for(i=1;i<=nrCifre;i++)
    for(j=1;j<=nrCifre;j++)
        if(i == j) a[i][j] = 0;
        else if(j > i) a[i][j] = mx;
        else a[i][j] = mn;
    for(i=1;i<=nrCifre;i++)
    {
        for(j=1;j<=nrCifre;j++)
            cout << a[i][j] << " ";
        cout << endl;
    }
    return 0;
}
```



### **Subiectul nr 15**

Scrieți un program care citește de la tastatură două numere naturale  $n$  și  $m$  ( $n < m$ ) și scrie în fișierul text ATESTAT.OUT toate numerele prime din intervalul deschis  $(n, m)$ .

Numerele se scriu în ordine crescătoare.

#### **Exemplu**

Pentru  $n = 87$  și  $m = 135$ , în fișierul ATESTAT.OUT se afișează: 89 97 101 103 107 109 113 127 131

```
#include <iostream>
#include <fstream>
using namespace std;
ofstream g("ATESTAT.OUT");
int n,m, k;
int prim (int x)
{
    for(int i=2; i<=x/2; i++)
        if(x%i==0) return 0;
    return 1;
}
int main()
{
    cin>>n>>m;
    for(k=n; k<=m; k++)
        if(prim(k)) g<<k<<" ";
    return 0;
}
```

### **Subiectul nr 16**

Să se scrie un program care citește un șir de caractere și afișează șirul oglindit din care lipsesc vocalele.

**Exemplu:** Pentru șirul citit : "Au inflorit trandafirii" se va afișa șirul: "rfdnrt trlfn"

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int i,n;
    char Text[256],voc[]="aeiouAEIOU";
    cin.get(Text,256);
    cin.get();
    n=strlen(Text);
    for(i=0;i<n;i++)
        if(strchr(voc,Text[i]))
            strcpy(Text+i,Text+i+1);
    strrev(Text);
    for(i=0;i<n;i++)
        if(strchr(voc,Text[i]))
            strcpy(Text+i,Text+i+1);
    cout<<Text;
    return 0;
}
```



### **Subiectul nr 17**

Se citesc de la tastatură  $n$  numere naturale, cu cel mult 9 cifre fiecare. Scrieți un program care afișează cifra care apare de cele mai multe ori în numerele citite. Dacă există mai multe astfel de cifre se vor afișa toate cifrele.

**Exemplu:** Pentru  $n=4$  și valorile 223, 431, 154, 452 se afișează 2 4 pentru că cifrele 2 și 4 apar de 3 ori în numerele citite.

```
#include <iostream>
using namespace std;
int main()
{ int n, fr[10]={0}, i, x, max=0;
  cout<<"n="; cin>>n;
  for(i=1; i<=n; i++)
  { cout<<"x="; cin>>x;
    while(x)
    { fr[x%10]++;
      if(fr[x%10]>max)
        max=fr[x%10];
      x/=10;
    }
  }
  for(i=0; i<10; i++)
    if(fr[i]==max)
      cout<<i<<" ";
  return 0;
}
```

### **Subiectul nr 18**

Scrieți un program care citește elementele unui vector  $v$  de la tastatură sub forma: dimensiunea vectorului  $n$  și apoi elementele sale, numere naturale. Să se inverseze elementele în vector fără a utiliza un vector intermediar, și să se afișeze vectorul astfel obținut.

#### **Exemplu**

Pentru  $n = 5$  și  $v = \{2, 9, 5, 7, 19\}$ , se afișează: 19 7 5 9 2

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



### **Subiectul nr 19**

Să se scrie un program care citește un șir de caractere și afișează câte cifre sunt în text, precum și numărul semnelor de punctuație din acel text(?!,.).

**Exemplu:** Maine, după ora 16, mergem în parc?

Se va afișa: număr cifre:2

Număr semne de punctuație:3

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{ char Text[256], cif[]="0123456789", sep[]="?!.,.";
  int i, NrCifre=0, NrSemne=0;
  cin.get(Text, 256);
  cin.get();
  for(i=0; i<strlen(Text); i++)
  { if(strchr(cif, Text[i])) NrCifre++;
    if(strchr(sep, Text[i])) NrSemne++;
  }
  cout<<"Numar cifre : "<<NrCifre<<endl;
  cout<<"Numar semen de punctuatie : "<<NrSemne;
  return 0;}
```

### **Subiectul nr 20**

Să se construiască o matrice cu n linii și n coloane ce memorează primii **nxn** termeni din șirul lui Fibonacci. (șirul lui Fibonacci :1,1,2,3,5,8,13,21,etc.)

**Exemplu:** n=3 se va afișa matricea :

1	1	2
3	5	8
13	21	34

```
#include<iostream>
using namespace std;
int main(){
  int n, i, j, a=1, b=1, c, x[100][100];
  cout<<"n=";cin >> n;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      if(i==1 && j==1) x[1][1] = 1;
      else if(i==1 && j==2) x[1][2] = 1;
      else{
        c = a + b;
        a = b;
        b = c;
        x[i][j] = b;
      }
  for(i=1;i<=n;i++)
  { for(j=1;j<=n;j++)
    cout << x[i][j] << " ";
    cout << endl;
  }
  return 0;}
```



### Subiectul nr 21

Scrieți un program care sortează crescător elementele unui vector de numere întregi  $v$  citit de la tastatură folosind una din metodele de sortare învățate. Vectorul se citește sub forma: dimensiunea vectorului  $n$  și apoi elementele sale. Vectorul sortat se va afișa pe ecran.

**Exemplu** Pentru  $n = 6$  și  $v = \{5, 3, 23, 11, 9, 6\}$ , se afișează: 3 5 6 9 11 23

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

### Subiectul nr 22

Fișierul `numere.in` conține pe prima linie cel mult un milion de numere naturale în ordine crescătoare dintre care cel puțin un număr care este o putere a lui 3. Scrieți un program care scrie în fișierul `numere.out` toate numerele ce sunt puteri ale lui 3.

**Exemplu**

numere.in	numere.out
27 43 54 81 452 729	27 81 729

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int x,aux;
    ifstream f("numere.in");
    ofstream g("numere.out");
    while(!f.eof() && f>>x)
    {
        aux=x;
        while(aux%3==0)
            aux=aux/3;
        if(aux==1) g<<x<<" ";
    }
    f.close(); g.close();
    return 0;
}
```





### **Subiectul nr 23**

Se citește de la tastatură un șir de  $n$  ( $n < 10$ ) numere naturale. Scrieți un program care verifică dacă numărul format din primele cifre ale elementelor din șir este palindrom (are aceeași valoare, citit de la stânga la dreapta și de la dreapta la stânga) și afișează numărul respectiv și un mesaj care să precizeze dacă el este sau nu palindrom.

**Exemplu** Pentru  $n=5$  și numerele 123, 435, 92, 4002, 10, se afișează: 14941 este palindrom, iar pentru  $n = 5$  și numerele 123, 435, 92, 4002, 30, se afișează: 14943 nu este palindrom

```
#include <iostream>
using namespace std;
int n,i, x, v[100];
long nr,inv,aux;
int main()
{   cin>>n;
    for(i=1; i<=n; i++) cin>>v[i];
    nr=0;
    for(i=1; i<=n; i++)
    {   x=v[i];
        while (x>9) x=x/10;
        nr=nr*10+x;
    }
    inv=0; aux=nr;
    while(aux)
    {   inv=inv*10+aux%10;
        aux=aux/10;
    }
    if(inv==nr) cout<<nr<<" este palindrom";
    else cout<<nr<<" nu este palindrom";
    return 0; }
```

### **Subiectul nr 24**

Să se scrie un program care citește un șir de caractere și înlocuiește toate aparițiile caracterului  $x$  citit de la tastatură cu un caracter  $y$ , citit de la tastatură.

**Exemplu:** Dacă se citesc următoarele informații în ordine :

Șirul : Mama duce o cutie mare, caracterul  $x='m'$  și caracterul  $y='t'$

Se va afișa: Tata duce o cutie tare

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{   char Text[256],x,y;
    int i;
    cin.get(Text,256); cin.get();
    cout<<"x=";cin>>x;
    cout<<"y=";cin>>y;
    for(i=0;i<=strlen(Text);i++)
        {if(Text[i]==x) Text[i]=y;
         if (Text[i]==toupper(x)) Text[i]=toupper(y);}
    cout<<Text;
    return 0;}
```



### Subiectul nr 25

Se citește de la tastatură un șir de  $n$  numere naturale. Scrieți un program care determină și afișează fracția subunitară ireductibilă care se poate forma din numărul minim și numărul maxim din șir.

**Exemplu** Pentru  $n = 4$  și numerele 3, 2, 5, 8, se afișează:  $1/4$

```
#include <iostream>
using namespace std;
int n,i, v[100], mini, maxi, a, b, r, d;
int main()
{
    cin>>n;
    for(i=1; i<=n; i++) cin>>v[i];
    mini=maxi=v[1];
    for(i=2; i<=n; i++)
    {
        if(v[i]>maxi) maxi=v[i];
        if(v[i]<mini) mini=v[i];
    }
    a=mini; b=maxi; r=a%b;
    while(r)
    {
        a=b;      b=r;      r=a%b;
    }
    d=b;
    cout<<mini/d<<"/"<<maxi/d;
    return 0;
}
```

### Subiectul nr 26

Fișierul ATESTAT.IN conține două linii. Pe prima linie sunt scrise trei numere naturale nenule, separate prin câte un spațiu:  $n$ , ( $5 < n < 100$ ),  $p \leq 9$ ,  $k \leq 9$  ( $k < p$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, reprezentând un șir de numere formate fiecare din cel mult 4 cifre.

Să se scrie un program în limbajul C++, care afișează pe ecran, în linie, separate printr-un spațiu, toate numerele din șir care împărțite la  $p$  dau restul  $k$ ;

**Exemplu:**

Fisierul ATESTAT.IN conține

7 5 3

2643 541 97 238 242 33 4682

Pe ecran se va afișa:

2643 238 33

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("atestat.in");
int n, i, p, k, v[100];
int main()
```

Exemplu: dacă se citește numărul 1359, matricea construită va fi cea alăturată.

9	5	3	1
9	5	3	1
9	5	3	1
9	5	3	1

```
{f>>n>>p>>k;
    for(i=1; i<=n; i++) f>>v[i];
    for(i=1; i<=n; i++)
        if(v[i]%p==k) cout<<v[i]<<" ";
    return 0;}
```



**Subiectul nr 27**

Se citește de la tastatură un tablou bidimensional cu n linii (numerotate de la 1 la n) și m coloane (numerotate de la 1 la m). Afișați separate prin spațiu liniile care conțin cel puțin un element cu valoarea 0.

**Exemplu:**

Pentru n=3 și m=4 și elementele tabloului bidimensional :

```
1 0 3 5
8 3 7 6
2 0 5 0
```

Se va afișa 1 3

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

**Subiectul nr 28**

Fișierul ATESTAT.IN conține două linii. Pe prima linie este scris numărul n (n<100). Pe cea de-a doua linie a fișierului sunt scrise n numere naturale separate prin câte un spațiu, reprezentând un șir de numere formate fiecare din cel mult 4 cifre.

Să se scrie un program în limbajul C++, care prin intermediul unui subprogram, afișează pe ecran, separate printr-un spațiu, numărul cifrelor pare din scrierea numerelor din șirul dat;

**Exemplu:**

Fisierul ATESTAT.IN conține

```
7
2643 541 97 238 242 33 4682
```

Pe ecran se va afișa: 13

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("atestat.in");
int n, i, nr, aux, v[100];
int main()
{ f>>n;
  for(i=1; i<=n; i++) f>>v[i];
  nr=0;
  for(i=1; i<=n; i++)
  { aux=v[i];
    while(aux)
      { if(aux%2==0) nr++;  aux=aux/10;  }
  }
  cout<<nr;
  return 0; }
```



### Subiectul nr 29

Fișierul ATESTAT.IN conține două linii. Pe prima linie este scris numărul  $n$  ( $n < 100$ ). Pe cea de-a doua linie a fișierului sunt scrise  $n$  numere naturale separate prin câte un spațiu, reprezentând un șir de numere formate fiecare din cel mult 4 cifre.

Să se scrie un program în limbajul C++, care verifică dacă toate elementele din șir sunt distincte două câte două și afișează în fișierul ATESTAT.OUT un mesaj corespunzător: "ELEMENTELE SUNT DISTINCTE" sau "ELEMENTELE NU SUNT DISTINCTE" - în caz contrar.

#### Exemplu:

Fișierul ATESTAT.IN conține

```
7
2643 541 97 238 242 33 4682
```

Fișierul ATESTAT.OUT conține

ELEMENTELE SUNT DISTINCTE

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("atestat.in");
ofstream g("atestat.out");
int n, i, j, ok, v[100];
int main()
{
    f >> n;
    for(i=1; i<=n; i++)
        f >> v[i];
    ok=1;
    for(i=1; i<=n-1; i++)
        for(j=i+1; j<=n; j++)
            if(v[i]==v[j])
                ok=0;
    if(ok)
        g << "ELEMENTELE SUNT DISTINCTE";
    else
        g << "ELEMENTELE NU SUNT DISTINCTE";
    return 0;
}
```

### Subiectul nr 30

Se citește un șir de numere naturale până la citirea lui 0 (care nu face parte din șir). Să se calculeze media aritmetică a numerelor de 3 cifre din șir care au cel puțin o cifră egală cu  $k$ , citit de la tastatură.

#### Exemplu:

Pentru  $k=5$

Pentru șirul 2, 751, 215, 210, 2157, 85, 550, 620, 0

Media aritmetică calculată este 505.3

```
#include <iostream>
#include <iomanip>
using namespace std;
int contor(int x, int k)
{ while(x>0)
    { if(k==x%10) return 1; x/=10; } return 0;
}
int main()
{ int n, s=0, k, i=0;
  cout << "k="; cin >> k;
  do{ cout << "n="; cin >> n;
    if(100<=n && n<1000 && contor(n,k)==1)
        { s+=n; i++; }
    } while(n!=0);
  cout << fixed << setprecision(1) << (float)s/i;
  return 0;
}
```



### **Subiectul nr 31**

Se citește de la tastatură un tablou bidimensional cu  $n$  linii și  $m$  coloane. Afișați cel mai mic element de pe fiecare coloană a tabloului.

#### **Exemplu:**

Pentru  $n=3$  și  $m=4$  se citesc elementele tabloului bidimensional :

7 2 2 4

3 9 7 6

1 0 5 8

Se va afișa 1 0 2 4

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

### **Subiectul nr 32**

Se citesc elementele unei matrice cu  $n$  linii și  $m$  coloane ( $1 < m, n < 6$ ), numere întregi, precum și o valoare întreagă  $k$ . Să se scrie un program care afișează liniile ce conțin valoarea  $k$ . Se va afișa un mesaj în situația în care nici o linie nu conține valoarea  $k$ .

#### **Exemplu:**

Pentru  $n=3$ ,  $m=4$ ,  $k=5$  se citesc elementele tabloului bidimensional

1 0 2 5

0 0 7 6

1 0 5 0

Se va afișa 1 3

```
#include <iostream>
using namespace std;
int main()
{ int k,i,j,n,m,ok2=0,ok1=0, Mat[100][100];
  cout<<"n=";cin>>n;      cout<<"m=";cin>>m;      cout<<"k=";cin>>k;
  for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
      { cout<<"Mat["<<i<<"["<<j<<"=";      cin>>Mat[i][j]; }
  for(i=1;i<=n;i++)
  {   for(j=1;j<=m;j++)
      if(Mat[i][j]==k)   { ok2=1; ok1=1; }
      if(ok2==1){ cout<<i<<" ";   ok2=0; }
  }
  if(ok1==0)      cout<<"Nu exista valoarea k in matrice";
  return 0;}
```



### Subiectul nr 33

Scrieți un program care afișează toate numerele naturale, mai mari decât 10 și mai mici decât o valoare data  $n$ ,  $n \leq 2.000.000$ , formate doar din cifre identice.

**Exemplu:** pentru  $n=195$  se afișează : 11 , 22 , 33 , 44 , 55 , 66 , 77 , 88 , 99 , 111

```
#include <iostream>
using namespace std;
int main()
{ int n,uc,i,ok,x;
  cout<<"n=";cin>>n;
  for(i=10;i<=n;i++)
    { x=i;ok=1; uc=x%10;
      while(x && ok)
        { if(uc!=x%10) ok=0; x/=10;
          }
      if(ok) cout<<i<<" ";
    }
  return 0; }
```

### Subiectul nr 34

Se introduc de la tastatură  $n$  ( $n \leq 10000$ ) numere naturale.

Să se construiască cu numerele citite un tablou bidimensional având  $m$  linii și  $p$  coloane. În cazul în care  $n < m \cdot p$  tabloul bidimensional se va completa cu zerouri. Să se afișeze tabloul astfel construit.

**Exemplu:**

Pentru  $n=10$ ,  $m=4$ ,  $p=3$  și numerele : 2,5, 13, 8, 45,9, 12, 5, 8, 17 se va afișa:

2 5 13

8 45 9

12 5 8

17 0 0

```
#include <iostream>
using namespace std;
int main()
{ int i,j,n,m,k,p,nr;
  cout<<"n=";cin>>n;
  cout<<"m=";cin>>m;
  cout<<"p=";cin>>p;
  int Mat[100][100];
  cout<<"introduceti numerele:";
  for(j=1;j<=m;j++)
    for(k=1;k<=p;k++)
      Mat[j][k]=0;
  j=1; k=1;
  for(i=1;i<=n;i++)
    { cin>>nr;
      if(k<=p)
        Mat[j][k++]=nr;
      else{j++;k=1;Mat[j][k]=nr;}
    }
  for(i=1;i<=m;i++){
    for(j=1;j<=p;j++)
      cout<<Mat[i][j]<<" ";
    cout<<endl;
  }
  return 0; }
```



### Subiectul nr 35

Se citesc mai multe numere întregi din fișierul ATESTAT.IN . Se cere să se afișeze cele mai mari două numere prime din fișier. Dacă în fișier nu sunt două numere cu această proprietate se va afișa un mesaj corespunzător

**Exemplu:** pentru fișierul cu numerele 23 56 7777 25 22 10 1012 54453 7 131 1171 9 293 31 67 313 se va afișa 1171 și 313.

```
#include<iostream>
#include<fstream>
using namespace std;
int x, ok, a, b, d;
int main()
{ ifstream f("date.in");
  while(f>>x)
  { ok=1;
    for(d=2;d<=x/2;d++)
      if(x%d==0) ok=0;
    if(ok==1 && x>a) { b=a; a=x;}
    else if(ok==1 && x>b) b=x;
  }
  if(b!=0) cout<<a<<" "<<b;
  else cout<<"Numere insuficiente";
  return 0;}
```

### Subiectul nr 36

Se citesc de la tastatură elementele unei matrice cu  $n \times n$  numere întregi , $n \leq 8$ . Să se numere acele valori din matrice care au suma ultimelor două cifre un număr impar.

**Exemplu:**

Pentru  $n=3$  se citesc elementele matricei :

```
24 5
7 9 11
8 71 16
```

Se va afișa: 5

```
#include <iostream>
using namespace std;
int main()
{ int i,j,n,m,c,k,nr,Mat[10][10];
  cout<<"n=";cin>>n;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      { cout<<"Mat["<<i<<"]["<<j<<"]="; cin>>Mat[i][j]; }
  nr=0;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      if(Mat[i][j]<10 && Mat[i][j]==1) nr++;
      else { c=Mat[i][j]%10;
             k=Mat[i][j]/10;
             k=k%10;
             if((c+k)%2==1)nr++;
           }
  cout<<nr; return 0;}
```





### Subiectul nr 37

Se citește un șir de caractere format din litere mici, litere mari și spații. Să se afișeze litera/literele mică/mici care apar/e de cele mai multe ori și de câte ori apare.

#### Exemplu:

Dacă se citește șirul *Andrei, Gabi și Bogdan joaca fotbal in curtea scolii* se va afișa *a i 6*.

```
#include<iostream>
#include<cstring>
using namespace std;
char s[256];
int i, v[26], maxi;
int main()
{ cout<<"Introduceti un text";
  cin.get(s, 256);
  for(i=0;i<strlen(s);i++)
    if(s[i]>='a' && s[i]<='z')
      v[s[i]-'a']++;
  for(i=0;i<26;i++)
    if(v[i]>maxi)
      maxi=v[i];
  for(i=0;i<26;i++)
    if(v[i]==maxi)
      cout<<(char) (i+'a')<<" ";
  cout<<maxi;
  return 0;}
```

### Subiectul nr 38

Se consideră o matrice oarecare de dimensiune **m**x**n** cu elemente numere naturale. Verificați dacă **ultima cifră** a produsului tuturor elementelor din matrice este egală cu **ultima cifră** a sumei elementelor din matrice.

$$A = \begin{pmatrix} 4 & 11 & 8 \\ 12 & 7 & 25 \end{pmatrix}$$

**Exemplu:**1) se citește: m=2 și n=3 și matricea

S=67 P=739200 se va afișa Nu

$$A = \begin{pmatrix} 0 & 2 & 2 \\ 2 & 2 & 2 \end{pmatrix}$$

2) se citește: m=2 și n=3 și matricea

S=10 P=0 se va afișa Da

```
#include <iostream>
using namespace std;
int main()
{ int i,j,n,m,P=1,S=0;
  int Mat[10][10];
  cout<<"m=";cin>>m;
  cout<<"n=";cin>>n;
  for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
      {cout<<"Mat["<<i<<"["<<j<<"]="; cin>>Mat[i][j];}
  for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
      {P=P*Mat[i][j]; S=S+Mat[i][j]; }
  if(P%10==S%10) cout<<"Da";
  else cout<<"Nu";
  return 0;}
```



### **Subiectul nr 39**

Se citește un șir de caractere format din litere mici, litere mari și spații. Să se afișeze perechile de vocale care apar în text și numărul perechilor.

#### **Exemplu:**

Dacă se citește șirul *Ioan, Gabi și Bogdan joaca fotbal in curtea spatiosa a scolii* se va afișa *Io, oa, oa, ea, io, oa, ii* 7.

```
#include<iostream>
#include<cstring>
using namespace std;
char s[256], voc[]="aeiouAEIOU";
int i, k;
int main()
{   cout<<"Dati sirul de caractere";
    cin.get(s, 256);
    for(i=0;i<strlen(s)-1;i++)
        if(strchr(voc, s[i])!=0 && strchr(voc, s[i+1])!=0)
        {   cout<<s[i]<<s[i+1]<<" ";
            k++;
        }
    cout<<k;    return 0;
}
```

### **Subiectul nr 40**

Fiind dată o matrice patratică de dimensiune **nxn**, să se afișeze în fișierul MATRICE.TXT matricea, fără elementele de pe cele două diagonale ale matricei și să se calculeze media aritmetică a elementelor pozitive din matrice. Media aritmetică se va afișa pe ultima linie a fișierului de ieșire.

#### **Exemplu:**n=3

1 2 3

4 5 6

7 8 9

MATRICE.TXT

2

4 6

8

Media numerelor pozitive este egala cu 5.

```
#include<iostream>
#include<fstream>
using namespace std;
int main(){
    ofstream f("matrice.txt");
    int n, i, a[100][100], j, suma=0, numerePozitive=0;
    cout<<"n=";cin>> n;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            cin >> a[i][j];
    for(i=1;i<=n;i++)
        {   for(j=1;j<=n;j++)
            {   if(i==j || i+j==n+1) cout << " ";
                else f << a[i][j] <<" ";
            }
            f << endl;
        }
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
            if(a[i][j] > 0) { suma = suma + a[i][j];
                            numerePozitive++;}
    f << "Media numerelor pozitive este egala cu " << suma/numerePozitive
    << ".";
    return 0;}
```



## Matrice rară

O matrice  $A(n,m)$  cu elemente numere întregi, se numește **rară** dacă majoritatea elementelor sale sunt egale cu zero. O matrice rară, având  $k$  elemente nenule, poate fi memorată folosind un șir  $X$  conținând  $k$  triplet de forma (linie, coloana, valoare). Elementele se memorează în ordine lexicografică (crescător) după linie și coloană.

De exemplu, pentru  $n=m=3$ , matricea  $A$

0	5	2
0	2	0
2	0	3

Se memorează sub forma șirului  $X = \{(1,2,5), (1,3,2), (2,2,2), (3,1,2), (3,3,3)\}$ .

Să se scrie un program care citește de la tastatură valorile  $n, m$  și două matrici rare  $A(n,m)$  și  $B(n,m)$ , calculează sub forma unei matrici rare  $C(n,m)$ , suma celor două matrici și afișează rezultatul sub forma unui tablou bidimensional.

```
#include <iostream>
using namespace std;
// definim o inregistrare cu campurile(linie, coloana, valoare)
//pentru memorarea elementelor nenule
struct element{
    int linie, coloana, valoare;
};

//definim o inregistrare pentru matricea rara
struct matrice_rara{
    int num_linii, num_coloane; // numar de linii, respectiv de coloane
    int num_elemente; // numarul de elemente nenule
    element v[100]; // sirul de elemente nenule
};

// Se verifica daca elementul e1 are (linie, coloana) "mai mica" decat
//(linie, coloana) a elementului e2
// returneaza 1 data e1 este inaintea lui e2

int mai_mic(element e1, element e2)
{
    if (e1.linie < e2.linie)
        return 1;
    if (e1.linie == e2.linie && e1.coloana < e2.coloana)
        return 1;
    return 0;
}
```



**/\* Inserarea unui element,x de forma (linie, coloana, valoare)**

in matricea rara reprezentata prin sirul de elemente ordonat crescator dupa (linie, coloana). \*/

**void inserare(matrice\_rara &m, element x)**

```
{
    if (m.num_elemente==0) // daca nu sunt elemente in sir
        m.v[0]=x; // x se adauga pe prima pozitie in sir
    else{
        int i=m.num_elemente;
        while (i>0 && mai_mic(x, m.v[i-1])){ // se cauta pozitia de inserare
            m.v[i]=m.v[i-1]; // elemente din sir care sunt "mai mari" decat x
                               // se deplaseaza cu o pozitie in dreapta
            i--;
        }
        m.v[i]=x; // s-a gasit pozitia corecta de inserare.
    }
    m.num_elemente++; // se incrementeaza numarul de elemente
}
```

**/\*Determinam elementul de pe o linie si coloana precizate.**

Se parcurge sirul de elemente si se cauta un element avand (linie, coloana) specificate.

Daca nu se va gasi un astfel de element, se va returna 0.\*/

**int determina\_element(matrice\_rara &m, int lin, int col)**

```
{
    for (int k = 0; k < m.num_elemente; k++)
        if (m.v[k].linie == lin && m.v[k].coloana == col)
            return m.v[k].valoare;
    return 0;
}
```

**/\*Pentru citirea matricei rare se citesc numarul de linii,**

numarul de coloane si apoi elementele pana la citirea tripletului (-1, -1, -1) \*/

**void citire(matrice\_rara &m)**

```
{
    element t;
    cout<<"numarul de linii = ";
    cin>>m.num_linii;
    cout<<"numarul de coloane = ";
    cin>>m.num_coloane;
    m.num_elemente = 0;
    cin >> t.linie >> t.coloana >> t.valoare;
    while (!(t.linie == -1 && t.coloana == -1 && t.valoare == -1)){
        if (determina_element(m, t.linie, t.coloana) == 0)
            // daca elementul nu a fost inserat deja
            inserare(m, t);
        cin >> t.linie >> t.coloana >> t.valoare;
    }
}
```



### **/\*Suma a doua matrici rare**

Deoarece sirurile de elemente corespunzatoare celor doua matrici a, respectiv b sunt ordonate, pentru determinarea elementelor din matricea suma ( $c=a+b$ ) se foloseste INTERCLASAREA.\*/

**void suma(matrice\_rara &a, matrice\_rara &b, matrice\_rara &c)**

```
{
    c.num_coloane=a.num_coloane;
    c.num_linii=a.num_linii;
    c.num_elemente=0;

    int i=0;    // indicele curent in sirul corespunzator matricii a
    int j=0;    // indicele curent in sirul corespunzator matricii b
    while (i < a.num_elemente && j < b.num_elemente){
        if (mai_mic(a.v[i], b.v[j])) // se adauga in matricea rara c elementul "mai mic"
            c.v[c.num_elemente++]=a.v[i++];
        else
            if (mai_mic(b.v[j], a.v[i]))
                c.v[c.num_elemente++]=b.v[j++];
            else{
                int s=a.v[i].valoare + b.v[j].valoare;
                if (s != 0){ //adaugarea in matricea rara se face doar daca suma este nenula
                    c.v[c.num_elemente]=a.v[i];
                    c.v[c.num_elemente++].valoare=s;
                }
                i++;
                j++;
            }
    }

    // se copiaza in c elemente ramase in a sau in b
    while (i < a.num_elemente)
        c.v[c.num_elemente++] = a.v[i++];
    while (j < b.num_elemente)
        c.v[c.num_elemente++] = b.v[j++];
}
```

// Se tipareste matricea rara sub forma unui tablou bidimensional

**void tipareste(matrice\_rara &m)**

```
{
    for (int i = 1; i <= m.num_linii; i++){
        for (int j = 1; j <= m.num_coloane; j++){
            cout << determina_element(m, i, j) << " ";
        }
        cout << endl;
    }
}
```

**int main()**

```
{
    matrice_rara a, b, c;
    citire(a);
    citire(b);
    suma(a, b, c);
    tipareste(c);
    return 0;
}
```

## NUMERE MAGICE

Se consideră două numere naturale  $p$  și  $q$  ( $2 \leq p \leq 10$ ,  $2 \leq q \leq 10$ ). Un număr natural se numește magic dacă mulțimea cifrelor utilizate în scrierea lui în sistemul de numerație având baza  $p$  este identică cu mulțimea cifrelor folosite în scrierea lui în sistemul de numerație având baza  $q$ .

De exemplu, pentru  $p = 9$  și  $q = 7$ ,  $(31)_{10}$  este număr magic pentru că  $(34)_9 = (43)_7$ , iar pentru  $p = 3$  și  $q = 9$ ,  $(9)_{10}$  este număr magic pentru că  $(100)_3 = (10)_9$ .

Scrieți un subalgoritm care, pentru două baze  $p$  și  $q$  date determină șirul  $x$  al tuturor numerelor magice strict mai mari ca 0 și strict mai mici decât un număr natural  $n$  dat ( $1 < n \leq 10\,000$ ). Parametrii de intrare ai subalgoritmului sunt  $p$  și  $q$  (cele două baze) și valoarea  $n$ . Parametrii de ieșire vor fi șirul  $x$  și lungimea  $k$  a șirului  $x$ . Exemplu: dacă  $p = 9$ ,  $q = 7$  și  $n = 500$ , șirul  $x$  va avea  $k = 11$  elemente: (1, 2, 3, 4, 5, 6, 31, 99, 198, 248, 297).

```
#include <iostream>
using namespace std;
void cifre(int n, int b, int v[])
{ /*construim un vector de aparitii pentru cifrele numarului
n in baza b*/
    for(int i=0;i<=9;i++)
        v[i]=0; //initializam vectorul cu 0
    while(n>0) //descompunem pe n in baza b
    {
        v[n%b]=1; //cifra lui n in baza b este n%b si marcam aparitia
        n=n/b;
    }
}

void numere_magice(int n, int p, int q, int &k, int v[])
{    int a[10], b[10], ok;
    k=0; //initial sunt 0 nr magice
    for(int x=1; x<=n; x++) //parcurgem intervalul [1,n]
    {
        cifre(x, p, a); //marchez cifrele lui x in baza p folosind tabloul a
        cifre(x, q, b); //marchez cifrele lui x in baza q folosind tabloul b
        ok=1;
        for(int i=0; i<=9; i++) //parcurem cifrele de la 0 la 9
            if(a[i]+b[i]==1) //daca apare doar in unul dintre tablouri
                ok=0; //x nu e magic
        if(ok==1) {k++; v[k]=x; } //x este magic si il adaugam in tabloul v
    }
}

int main()
{    int k, v[100], n, p, q;
    cin >> n >> p >> q;
    numere_magice(n, p, q, k, v);
    for(int i=1; i<=k; i++)
        cout << v[i] << " ";
    return 0;
}
```



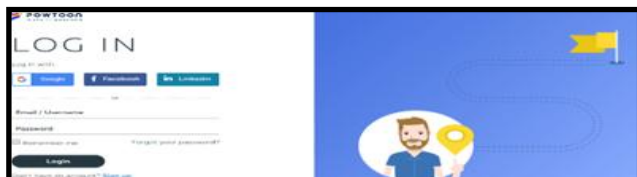
## CREAREA DE PREZENTĂRI CU RESURSE FREE – APLICAȚIA POWTOON

Prof. Maria Stan, Colegiul Tehnic „Radu Negru” Galați

**PowToon** este o aplicație on – line care permite să se creeze animații gratuite și profesionale, tutoriale video / mesaje și prezentări multimedia ca o alternativă la PowerPoint.

### Accesul la PowToon

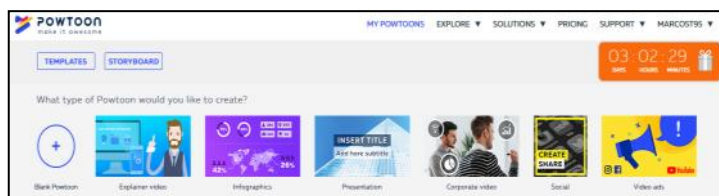
#### Pasul 1: logare



Conectare la <http://www.powtoon.com/>. Dacă doriți, vă puteți folosi contul de Google, de Facebook sau LinkedIn pentru a accesa PowToon.

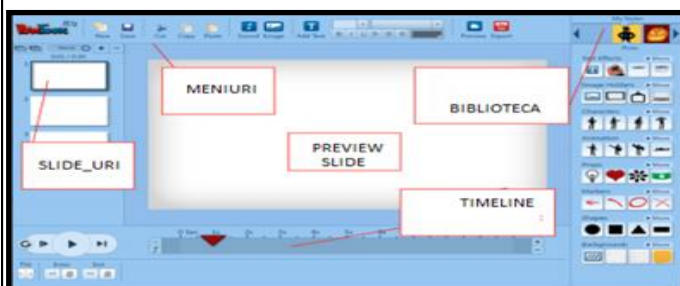
#### Pasul 2: Crearea un PowToon

Se poate crea un PowToon pornind de la un template sau să se conceapă direct o ”poveste a tablei” pe baza diaporizitelor predefinite.



#### Pasul 3 Cunoașterea Interfaței

Biblioteca PowToon este alcătuită din diferite categorii de text, obiecte (forme, marcare), elemente de recuzită și caractere care pot fi adăugate la prezentare. Se pot adăuga propriile imagini în biblioteca prin selectarea opțiunii "Image Holder" (image cadru).

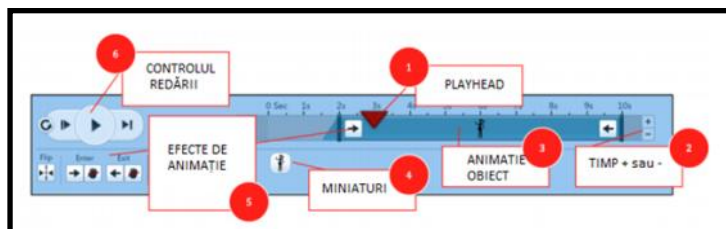


#### Utilizarea elementelor din Bibliotecă

Pentru a utiliza Biblioteca, selectați un stil din partea de sus a panoului. Se adaugă obiecte, elemente de recuzită și caractere, pur și simplu trageți / fixați pe diapozitiv. Obs. Unele articole din Biblioteca pot face parte din PowToon premium și nu sunt free.

#### Cronologia

*Cronologia PowToon- timeline-ul* - este utilizată pentru a adăuga sincronizarea animației către obiectele din prezentare.



1. *Playhead*: Playhead-ul indică la ce secundă sunteți în diapozitiv. Se poate glisa bara de derulare pe întreaga cronologie pentru a previzualiza animația pentru fiecare diapozitiv.



2. *Timp total de expunere - Total Slide Time*: Fiecare slide în mod prestabilit este atribuit 10 secunde. Acest lucru poate fi redus sau crescut până la maximum 20 de secunde.

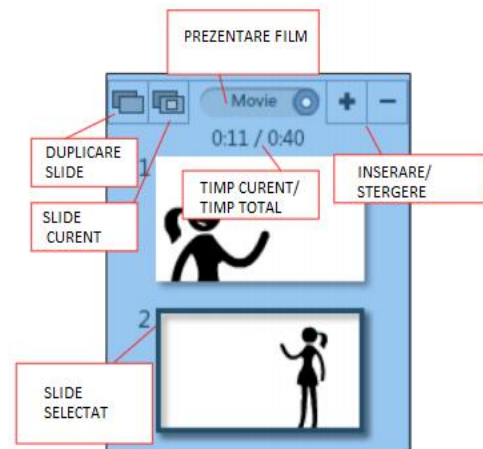
3. *Bara de animație a obiectului - Object Duration Bar*: Fiecare obiect are o bară de durată a timpului de animație.

Pentru a accesa această bară, selectați obiectul și reglați animația după cum doriți.

4. *Miniaturi/ obiecte - Object Thumbnails*: Toate obiectele din diapozitiv vor apărea sub cronologie. Faceți clic pe un obiect pentru ajustați efectul și durata obiectului.

5. *Efecte de obiecte (animație) - Object Effects (Animation)*: Obiectele pot fi setate la introducere sau ieșire din prezentare utilizând o varietate de opțiuni, inclusiv decolorare, direcționalitate etc. Fiecare obiect poate fi animat folosind aceste efecte ( se poate înclina un obiect din această zonă).

6. *Controlul redării - Playback Controls*: ajută la previzualizarea animației prezentărilor pe măsură ce se construiește. Există patru opțiuni de redare de la stânga la dreapta (a) redă toate diapozitivele de la început, (b) redare diapozitiv curent de la început, (c) redare de la diapozitivul existent și continuare și (d) redare diapozitiv curent și oprire.



## Panoul de diapozitive

Panoul de diapozitive conține diapozitivele pentru prezentare.

Se pot adăuga / elimina și duplica diapozitive (toate obiectele și calendarul). De asemenea, se poate clona diapozitivele dorite. Acest lucru ajută realizarea unei tranziții line de la diapozitiv la diapozitiv.

Prezentare / Mod film: Comutare între modul de filmare și prezentare în funcție de cum dorește.

Tranziții: pot fi adăugate tranziții între diapozitive accesând opțiunile de tranziție din panoul Bibliotecă.

Pentru a adăuga o tranziție, selectați diapozitivul și faceți clic pe dorit efect de tranziție. Tranziția va apărea sub valoarea selectată și numele efectului de tranziție va apărea între cele două diapozitive. Notă: Tranzițiile sunt o lungime standard și nu pot fi editate. Pentru a elimina o tranziție, selectați diapozitivul și din Opțiuni de tranziție din panoul Bibliotecă, selectați None (niciuna).

Layouts: includ layout-uri pentru diapozitive animații și elemente de bază ale temporizării. Se accesează opțiunile pentru aspectul diapozitivului dând clic pe diapozitiv și selectând fila opțiuni de aspect. In meniu vor apărea layout-uri.

Editați elementele individuale pe diapozitiv și cronometre pentru a ajusta efectele. Se poate să aplicați doar un aspect/opțiune pentru un slide o singură dată. Pentru a aplica un aspect nou, adăugați pur și simplu un diapozitiv nou și selectați o opțiune nouă de aspect.

## Meniurile

1. **SAVE - Salvare**: deși PowToon se salvează automat la fiecare 5 minute, este important să putem salva în orice moment

2. **NEW- Nou**: pentru a crea un "PowToon" nou, o preăentare nouă.

3. **Cut, Copy, Paste**: tăiați, copiați și inserați diapozitivele, obiectele din prezentarea.

4. **SOUND- Sunet**: adăugați un fișier de sunet (fie importați, înregistrați sau utilizați o piesă de la PowToon).

5. **IMAGE- Imagine**: Importați imagini de pe computer sau de pe Web.

6. **Text**: adăugați text la o opțiune de diapozitive și formate text.

7. **PREVIEW- Previzualizare (Share / Link, Embed)**: Selectați previzualizarea pentru a revizui PowToon și a vă distribui prin intermediul unei adrese URL link-ul, site-ul social media sau încorpora într-un site web.

8. **EXPORTING -Export (YouTube)**: distribuiți direct PowToon pe YouTube sau exportați ca fișier video. Conturile gratuite includ sigla PowToon.

## FLUXUL DE LUCRU DE BAZĂ

### Pasul 1: Storyboard

Este recomandat să vă planificați prezentarea folosind o poveste care vă conturează ideile vizuale și expozițiunea. Este cea mai eficientă abordare pentru animație.

### Pasul 2: Adăugați Voiceover / Muzică

Este mult mai ușor să se adauge mai întâi vocalul, apoi să se adauge animații. De ce? Acesta oferă calendarul/planificarea, simplificând adăugarea de diapozitive și animații.

Adăugarea sunetului

Din panoul meniului, selectați Sunet sau selectați pictograma audio din Cronologie.

1. Import sau Înregistrare voce
2. Import sau adăugare melodie - Obs.: Unele tonuri nu sunt free.
3. Reglați sunetul cu comenzile specifice
4. Previzualizare / redare
5. Expunere prezentare

### Pasul 3: Adăugarea diapozitivelor

Utilizând panoul de diapozitive adăugați diapozitive la prezentare pe baza expunerii dorite.

### Pasul 4: Adăugați obiecte

Trageți și plasați obiecte din panoul Bibliotecii PowToon pe diapozitivele (inclusiv personaje, text, obiecte și elemente de recuzită).

Indicație: Se poate amplasa obiecte în straturi unul peste celălalt și să se utilizeze animație pentru a introduce obiecte strategic, pentru a crea un sentiment de mișcare dinamică și interacțiune cu desene animate.

### Gestionarea obiectelor

Gestionarea obiectelor în PowToon este similară cu MS Power point.

- Clic pe stânga pe elementul de selectat și oricare dintre ele țineți și mutați pe scenă cu ajutorul mouse-ului sau utilizați tastele săgeți.
- Utilizați mânerul obiectului pentru a redimensiona obiect.
- Clic dreapta pentru administrarea mai multor obiecte opțiuni cum ar fi aranjament (spate, față), orientare (flip), etc.

### Pasul 5: Animația prezentării

Obiecte - Din Timeline utilizați Bara de durată pentru a anima fiecare obiect pe diapozitiv.

1. Selectați obiectul.
2. Alegeți un stil de efect pentru obiect intrare / ieșire.
3. Utilizați glisoarele la ajustați durata animația obiectului din diapozitiv.
4. Adăugați / Îndepărtați sincronizarea generală a diapozitivelor după cum este necesar (diapozitivele pot avea o lungime de până la 20 de secunde).
- 5: Previzualizare

Utilizați comenzile de redare SAU selectați Previzualizare din panoul meniului pentru a revedea videoclipul. NOTĂ:

Nu uitați să salvați pe parcursul procesului de creare a prezentării. Caracteristica de salvare este localizată în meniul specific.

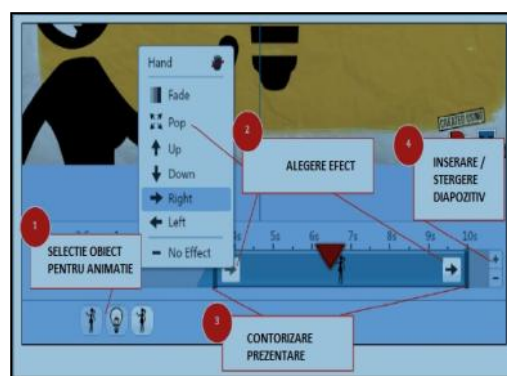
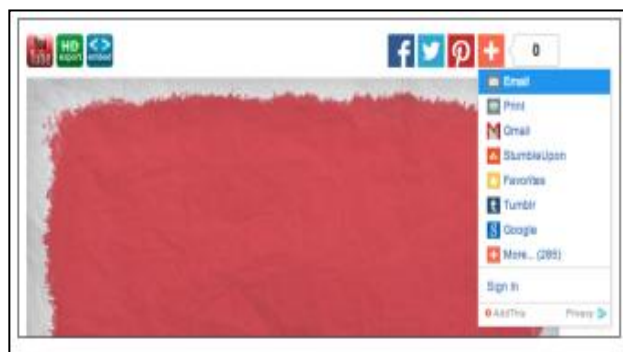
### Pasul 6: Publicare

Publicarea PowToon

- a: Distribuți prin link sau încorporați
- b. - Selectați Previzualizare din meniu
- Alegeți unul dintre partajare din opțiunile de previzualizare.

### Exportul pe YouTube / Altele

1. Selectați Export din meniu
2. În caseta de dialog selectați opțiunea de ieșire și Adăugați un cont YouTube. GRATUIT din Contul PowToon se poate exporta până la 30 de videoclipuri la rezoluție medie.



## Bibliografie

- <http://support.powtoon.com/knowledgebase/articles/all>
- <https://tinyurl.com/POWTOONebook>
- <http://www.powtoon.com/blog/>
- <http://www.powtoon.com/blog/russeexplains/>
- <http://www.powtoon.com/blog/justamminuteexplainers/>
- <https://tinyurl.com/POWTOONPres>





Fundația EOS a anunțat proiectul [Informatica 365](#), în parteneriat cu Microsoft. Ideea de la care a pornit a fost că informatica, dar și TIC, trebuie abordate cu instrumente moderne.

Proiectul îi vizează pe profesorii de gimnaziu care predau Informatică și TIC, însă beneficiarii finali sunt elevii. Prin acest program, aceștia pot dobândi cunoștințe de programare prin intermediul Minecraft pentru Educație și prin cursurile de [Programare Creativă](#) prin Jocuri și Aplicații.

Prin acest program, România se apropie de alte țări din Europa, unde informatica este materie de bază și sunt deja folosite instrumente noi. În caz că ți-e străină materia în România, încă sunt școli (foarte multe) unde C++ sau Pascal sunt limbaje de programare în care elevii ar trebui să învețe. Sistemul educațional nu va reuși niciodată să țină pasul cu cererea din piață, dar se poate apropia. Pentru asta, este nevoie și de profesori instruiți corespunzător, ceea ce programul Informatica 365 speră să ofere.

Proiectul sprijină inițiativa Ministerului Educației de a introduce programarea într-un stadiu timpuriu al ciclului educațional. Profesorii vor putea urma cursurile de Minecraft pentru Educație și Programare Creativă prin Jocuri și Aplicații.

Minecraft pentru Educație este o versiune educațională a jocului [Minecraft](#) (deținut de Microsoft) și reprezintă un instrument didactic interactiv. Cu ajutorul acestuia, copiii vor învăța termeni de programare și vor realiza proiecte individuale și în echipe. Astfel ei vor putea înțelege noțiuni de cod, de geometrie și vor fi inițiați în domenii precum arheologie, arhitectură sau morfologie.

Programarea Creativă prin Jocuri și Aplicații este o suită de cursuri dezvoltate în tehnologia TouchDevelop. Cursul este menit să stimuleze creativitatea și curiozitatea pentru informatică. Prin joacă, elevii vor învăța să scrie coduri și să asimileze termeni din vocabularul programării software.

Investiția este de 200.000 de dolari și cuprinde acreditarea celor două cursuri menționate mai sus și traducerea resurselor didactice în limba română. Prin intermediul programului vor fi instruiți 3.300 de profesori, în special din zonele dezavantajate și din mediul rural.

**In cazul in care copiii vorbesc non-stop despre smaralde, tarnacoape sau liane, instrumente indispensabile in jocul Minecraft, este foarte posibil ca parintii lor sa aiba la indemana o metoda eficienta de a transforma dependenta de jocuri video intr-o lectie de despre bani.**

Minecraft, un joc video fenomen, cu peste 100 de milioane de utilizatori, pe care [Microsoft](#) tocmai l-a cumparat pentru 2,5 miliarde dolari de la producatorul sau, Mojang AB, le poate da o mana de ajutor, apreciaza [Reuters](#). In Minecraft, jucatorii isi pot construi propriile lor lumi pe console, smartphone-uri, tablete sau PC-uri utilizand piese virtuale care seamana cu cele Lego. Si, spre deosebire jocurile video "shoot-'em-up" - pe care copiii, de obicei, le adora - Minecraft ii poate invata multe despre bani.

Este vorba "despre barter, despre valoare, despre cum sa isi protejeze lucrurile", spune Hank Mulvihill, un consilier financiar in Richardson, Texas.

"Copiii invata despre bani intr-o multime de feluri diferite in Minecraft", a adaugat Joel Levin, co-fondator al TeacherGaming din Manhattan, o firma care sustine ca invatatorii ar trebui sa utilizeze jocurile video ca instrumente de predare. "Exista moneda de baza, cum ar fi smaraldele, care este cautata si cu care pot face tranzactii," explica Levin. "Exista cursuri de schimb, pentru ca anumite elemente sunt mai valoroase decat altele. Apoi, jucatorii trebuie decida daca vor cheltui banii imediat, sau ii vor depune pentru a obtine mai mult, mai tarziu. Acestea sunt similare cu deciziile financiare pe care oameni le iau zilnic in lumea reala".

"Aici, jucatorii infiinteaza economii reale", spune Levin. "Ei pot decide care este moneda de schimb. Sau unii pot juca rolul unei banci, care ofera imprumuturi si cere dobanda".

Levin aminteste si lectiile prin care invatatorii pot prezenta element nou, pe care copiii nu il pot obtine imediat, fiind urmarita reactia acestora. "Cererea si oferta sunt in actiune", a mai adaugat acesta.

### **Cum se incepe totul de la zero**

Desigur, cei mai multi parinti au aflat despre Minecraft urmarind ce fac copii lor si incercand sa realizeze ce se intampla. Deci, sunt cateva lectii cheie in Minecraft.

La inceputul jocului, "nimeni nu-ti spune nimic, nu primesti instructiuni", spune Harvey Mulvihill, 19 ani, fiul lui Hank, ce joaca Minecraft impreuna cu cei doi frati. "Esti un strain intr-o tara straina si nu ai niciun indiciu despre cum sa obtii resurse"

Minecraft este un asa-numit joc "sandbox", in care jucatorii cutureiera o lume virtuala cu foarte putine limitari. Poate fi comparat cu traditionalul arhetip american din povestea lui Horatio Alger, unde, pornind de la nimic, eroul obtine succes. Jucatorii trebuie sa castige indemanari pe care apoi sa le foloseasca pentru a-si dezvolta o lume mai buna.

### **Cum sa protejezi ce ai castigat**

Ca si in viata reala, si in Minecraft se pot intampla multe nenorociri - moarte, jaf, atacuri fizice si dezastre de tot felul. Asa ca jucatorul trebuie sa se protejeze impotriva acestora.

De exemplu: o calatorie cu cat mai putin bagaj. "Nu este niciodata o idee buna ca cineva sa transporte obiecte de valoare", sfatuieste Patrick Mulvihill, 17 de ani. "Odata ce vei muri, lucrurile tale rama acolo unde s-a intamplat. Obiectele de valoare ar trebui sa fie pastrate in locuri sigure".

Insa jucatorii nu trebuie sa se asigure doar impotriva mortii personajului lor. In lumea virtuala sunt unele personaje - cunoscute sub numele de "griefer", potrivit lui Patrick - care intra online numai pentru a jefui alti jucatori.

### **Cum sa stabilesti relatiile cele mai bune**

In viata, conteaza foarte mult pe cine cunosti, iar in Minecraft nu este nicio diferenta. Relatiile cu persoanele potrivite pot face mai buna viata virtuala a jucatorului.

"Daca as fi inceput de la zero si nu as fi fost ajutat, as fi putut sapa dupa smaralde pentru totdeauna", a spus Dan Short, profesor asociat care predă stiinta mediului la Universitatea Robert Morris din Pittsburgh, care a scris o lucrare academica despre Minecraft. "Insa, pe un server cu alti copii, proprietarul acestuia poate da practic cat de multe smaralde sunt necesare. Copiii sunt mici regi".

Morala povestii? Conteaza reteaua, iar jucatorul care si-a creat reteaua potrivita, se poate dezvolta cat mai mult.

### **Cum iti poti valorifica indemanarile castigate**

Odata ce jucatorul s-a specializat in ceva anume, ar putea sa devina foarte cautat. Sean Mulvihill, in varsta de 15 ani, joaca Minecraft impreuna cu prietenii sai Jackson, Oscar si Wyatt. "Jackson este cel mai bun constructor, iar altii il platesc pentru a le construi case", a spus Sean, care s-a descris, cu modestie, ca fiind "de departe, cel mai bun fermier". Drept rezultat, alti jucatori ii propun afaceri, cum ar fi furnizarea de seminte, echipament sau aur, cu scopul de a face fermele mai eficiente si de a imparti profitul.

Indemanarile castigate in joc sunt utile si in lumea reala, mai spune Short. Unii - cum sunt TheBajanCanadian, Sky Does Minecraft sau Lewis & Simon - au devenit atat talentati in joc si atat de carismatici cu propriul lor public, incat au propriile lor canale, extraordinar de populare, pe YouTube.

"Castiga adepti pe YouTube, pot gazdui jocuri, iar unii copii platesc o taxa pentru a putea juca impreuna cu ei si pentru a fi in videoclipurile lor", a precizat Short. "Ei pot obtine venituri mari. Iar aceasta ar trebui sa fie cea mai buna lectie financiara".

Preluare articol Sursa: Business24

<https://playtech.ro/2018/programul-predare-informatica-romania/>  
prof.ing. Burlacu Cătălina, Liceul Teoretic Dunărea



## Colectivul de redacție:

- Redactor șef: CĂTĂLINA BURLACU – Liceul Teoretic „Dunărea”
- Redactor șef adjunct GEORGETA-IULIA BALACEA, C. N. “V. Alecsandri”
- Redactori principali profesorii:
  - o Inspector școlar: prof. Mușunoiu Novetschi Iuliana Monica
  - o Stan Maria, Colegiul Tehnic "Radu Negru"

## Redactori, profesorii:

- ⇒ Mirela Voinea, Cobzaru Luminița, Dău Liliana, Daniela Nistor—C. N. “Vasile Alecsandri”
- ⇒ Șoldan Florin—Palatul Copiilor
- ⇒ Neagu Violeta, Lucian Dimitrievici, Popescu Mădălina—C. N. “M. K ogălniceanu”
- ⇒ Bibicu Dorin, Zeld Jenica, Gavrila Iulia, Ionela Fotache, Tudor Veronica - L.T „Dunărea”
- ⇒ Joc Genia, Magdalena Grigore, Ilie Daniela C.N. “Al. I. Cuza”
- ⇒ Stratulat Daniela C. N. „C. NEGRI”
- ⇒ Moldovanu Simona—CTAT “D. Moțoc”
- ⇒ Penu Ionela—Școala Gimnazială nr. 2 & 12
- ⇒ Gheorghiu Carmen , Gheorghiu Bogdan—C.N. “Spiru Haret” Tecuci
- ⇒ Adriana Ciocănel—C.N. “Calistrat Hogaș” Tecuci

*Stimați colegi și elevi, vă invităm să colaborați la revista noastră. Responsabilitatea în legătură cu originalitatea și conținutul științific al articolelor, problemelor aparține în exclusivitate autorilor.*

*Felicitări elevilor și profesorilor care au înțeles că scopul acestei reviste este de a crea o relație elev-profesor fructuoasă. Așteptăm opiniile, problemele rezolvate și propuse pe adresa de mail a revistei noastre: [revistainformaticagalati@yahoo.com](mailto:revistainformaticagalati@yahoo.com).*

*Doneaza 2% din impozitul pe venit pentru Asociația Profesorilor de Informatică “Tudor Sorin”.*

**Cont bancar (IBAN) RO05BRDE180SV49067421800.**

**REDACTIA**



Coperta a fost realizată de FILIP LIDIA, C. N. “Vasile Alecsandri”







CERERE PRIVIND DESTINAȚIA SUMEI REPREZENTÂND  
PÂNĂ LA 2% DIN IMPOZITUL ANUAL ȘI DEDUCEREA  
CHELTUIELILOR EFECTUATE PENTRU ECONOMISIREA ÎN  
SISTEM COLECTIV PENTRU DOMENIUL LOCATIV

230

Anul:

I. DATE DE IDENTIFICARE A CONTRIBUABILULUI

Nume	Prezenta țară	Codul național	Cod numeric personal / Număr de identificare fiscală
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Județ/Sector	Localitate	Stradă	Număr
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Scara	Ap.	Cod poștal	Telefon
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
		Fax	E-mail
		<input type="text"/>	<input type="text"/>

II. DESTINAȚIA SUMEI REPREZENTÂND PÂNĂ LA 2% DIN IMPOZITUL ANUAL, POTRIVIT ART.57  
ALIN.(4) DIN LEGEA NR.571/2003

☐ 1. Bursa privată

Contract nr./data

Suma plătită

- lei -

Documente de plată nr./data

☒ 2. Susținerea unei entități nonprofit/unități de cult

Suma

- lei -

Denumirea entității nonprofit/unități de cult

ASOCIAȚIA PROFESORILOR DE INFORMATICĂ "TUDOR SORIN"

Codul de identificare fiscală  
a entității nonprofit/unități de cult

25790290

Cont bancar (IBAN) a entității nonprofit/unități de cult

RO05BRDE180SV49067421800

☐ III. DEDUCEREA CHELTUIELILOR EFECTUATE PENTRU ECONOMISIRE ÎN SISTEM COLECTIV PENTRU  
DOMENIUL LOCATIV DIN VENITURILE IMPOZABILE DIN SALARII, OBȚINUTE LA FUNCȚIA DE BAZĂ

Denumirea instituției de credit

Documente anexate

Suma plătită

- lei -

Sub sancțiunile aplicate faptelor de fals în acte publice, declar că datele înscrise în acest formular sunt corecte și complete.

Semnătură contribuabil

Data

Loc rezervat organului fiscal

Nr. înregistrare:

Data:

NOTĂ: Contribuabilii care solicită deducerea din veniturile impozabile din salarii, obținute la funcția de bază, a cheltuielilor efectuate pentru economisirea în sistem colectiv pentru domeniul locativ, vor anexa, în copie, formularul 210 - "Fișă fiscală", pentru veniturile realizate la funcția de bază.

Model de operator de date cu caracter personal 1067

Cod 14.12.04.12



CENTRUL NAȚIONAL ISSN  
BIBLIOTECA NAȚIONALĂ A ROMÂNIEI  
Bd. Unirii nr. 22, sect. 3, cod 030833  
București - ROMANIA  
Tel. +40 21 311.26.35  
Tel./Fax +40 21 314.24.34/1035  
e-mail: [issn@bibnat.ro](mailto:issn@bibnat.ro)



București, 17.04.2018

## Către Asociația Profesorilor de Informatică „Tudor Sorin” Galați

Vă anunțăm ca publicația pe care o editați a fost înregistrată și a primit codul de identificare ISSN, după cum urmează:

**Revista informatică (Im)puls++ (Online) =**  
**ISSN 2601 - 5951, ISSN-L 2067 - 9815**

Codul ISSN va fi utilizat conform instrucțiunilor conținute în anexa “ISSN – informații generale”, respectând Legea nr. 111/1995 republicată și Legea nr. 186/2003, privind promovarea culturii scrise.

Informații complete și actualizate despre ISSN și publicații seriale se găsesc și pe pagina web a Bibliotecii Naționale a României ([www.bibnat.ro](http://www.bibnat.ro))

Teodora Ușurelu  
Centrul Național ISSN

