

COLEGIUL NAȚIONAL DE INFORMATICĂ
PIATRA NEAMȚ

INFO ++

Nr. 2, MARTIE 2009

☞ CONCURSUL PROSOFT@NT

☞ PREGĂTIRE PENTRU
CONCURSURI

☞ CDȘ

☞ NOUȚĂȚI



CUPRINS

Argument	1
PROSOFT@NT – secțiunea programare	2
Problema cadouri	2
Problema sorcova	3
Problema joc	5
Problema numere	6
Problema garaj	7
Problema hipic	10
Profesor–elev–computer, expresia generală a mileniului III	11
Pregătire pentru olimpiada	12
Terenuri	12
Concursul de informatică aplicată	14
Model de subiect	14
Bacalaureat 2009	16
Model de subiect	16
ECDL	20
Permisul european de conducere a computerului	20
Platforma .NET	24
Platforma .NET	24
Tehnici bazate pe reprezentarea datelor	28
CDS	30
Interfețe grafice în JAVA	30
Noutăți	33
Mecatronica, tehnologia viitorului	33
ZOHO – suita Office online	34
Un nou sistem de operare – UBUNTU	34
Glume de informat	36

COLECTIVUL DE REDACȚIE

INFO ++

ISSN 2065 – 8184

Editor: Colegiul Național de
Informatică Piatra-Neamț

Redactori:

prof. Gabriela BLAGA, prof. Vasile DIACONU

Coperta:

prof. Gabriela BLAGA

Tehnoredactare computerizată:

Daniel DOROBANȚU

Ovidiu NECHITA

Adresa redacției:

Colegiul Național de
Informatică Piatra-Neamț,
Str. Mihai Viteazu, nr. 12
Tel./fax: 0233-227510
e-mail: cni_pn@yahoo.com

ARGUMENT

Societatea informațională produce schimbări în existența colectivă și individuală. Orice cetățean trebuie să înțeleagă rolul noilor tehnologii ale informației și să le folosească pentru sine și în folosul comunitar.

Sistemul educațional trebuie să formeze **o nouă cultură a informației și comunicării**, tot atât de necesară unui **cetățean european** ca și cultura civică.

Pentru a trăi, învăța și lucra cu succes într-o societate din ce în ce mai complexă și plină de informații, tinerii trebuie să folosească tehnologia în mod efektiv.

Revista de informatică **INFO++** promovează utilizarea calculatorului într-un mediu educativ sănătos pentru a-i ajuta pe tineri:

- ▶ Să devină utilizatori avizați de informație, capabili să caute informația, s-o analizeze, s-o evalueze și s-o recepteze critic;
- ▶ Să fie capabili să folosească un nou mediu de învățare bazat pe tehnologie, pentru a învăța, a comunica, a colabora, a produce și a-și dezvolta cunoștințele;
- ▶ Să fie capabili să rezolve probleme și să ia decizii;
- ▶ Să devină utilizatori efectivi și creativi ai instrumentelor productivității;
- ▶ Să devină cetățeni informați, responsabili și activi.

Revista **INFO++** se vrea a fi o reflectare a experiențelor multiple din domeniul informaticii din ultimul an, fiind produsul conlucrării dintre profesorii și elevii CNI, oferind în același timp elevului dornic de afirmare un bun instrument care să-l ghideze către succes.

Răsfoind paginile acesteia, atenția se focalizează către concursurile de informatică de anvergură. Astfel, sunt prezentate problemele și soluțiile date la Concursul Prosoft@NT – secțiunea programare (clasele IX-XII); modele de subiecte de la Concursul de Informatică Aplicată; un model de subiect de bacalaureat și rezolvarea acestuia; apoi sunt inserate motivațiile a doi elevi premianți la secțiunea soft educațional Prosoft@NT; ulterior se face o demonstrație de pregătire a unui olimpic la faza națională.

De asemenea, domniile profesori vin în sprijinul celor care doresc să obțină Permisul European de Conducere a Computerului, aducând lămuriri în modalitatea de concepere a modulelor și a subiectelor care atestă asimilarea cunoștințelor.

Elementele de programare superioară sunt dezvăluite în două articole denumite “Platforma .NET” și “Interfețe grafice în Java” care inițiază elevul în noutățile în domeniu.

Totodată, din dinamica perpetuă a informaticii, ochiul pasionat al elevului CNI selectează acele imagini pentru a le face cunoscute celorlalți: ”Mecatronica – tehnologia viitorului”, ”Zoho – suita office online”, ”Ubuntu – un nou sistem de operare”.

Pentru ca să se producă un minim echilibru între dificultatea pe care o presupune această disciplină și sprijinul jovial dintotdeauna românului, revista oferă în final ”Glume de informat” astfel încât tensiunea cerebrală se destinde treptat.

Prin urmare, revista se prezintă de la sine și se adresează atât elevilor capabili de performanță, cât și profesorilor care îi călăuzesc pe aceștia în împlinirea dezideratelor.

prof. Vasile DIACONU
director adjunct

PROBLEMA CADOURI (clasa a IX-a)

prof. Gheorghe Manolache

Moș Crăciun a anunțat la o conferință de presă și în acest an trebuie să distribuie daruri la toți copiii cuminiți iar de la bancă a fost informat că mai are în cont o sumă de bani pentru a cumpăra daruri care e rezultatul produsului a n numere naturale nenule cunoscute x_1, x_2, \dots, x_n . Dar **dorința moșului** este ca fiecare copil să primească daruri de aceeași valoare totală k, un număr natural, care ridicat la o puterea m cunoscută reprezintă exact suma totală de bani pentru daruri de care dispune moșul.

Cerință

Să se verifice dacă se poate îndeplini dorința moșului. În caz afirmativ să se afișeze și descompunerea în factori primi a numărului natural k ce reprezintă valoarea cadourilor primite de fiecare copil cuminte.

Date de intrare

În fișierul **cadou.in** se află pe prima linie **m**, pe linia a doua **n**, iar pe linia a treia numerele x_1, x_2, \dots, x_n separate între ele prin câte un spațiu.

Date de ieșire

În fișierul **cadou.out** se va scrie pe prima linie cifra 0, dacă dorința moșului nu se poate realiza pentru că valoarea cadourilor pentru fiecare copil nu este un număr natural, respectiv 1 dacă este un număr natural. Dacă valoarea expresiei este un număr natural pe următoarele linii se vor scrie perechi de forma **p e** (**p** este factor prim care apare în descompunere la puterea $e \geq 1$). Aceste perechi se vor scrie în ordine crescătoare după primul număr (adică **p**).

Restricții

- **n** – număr natural nenul <5000
- x_i – număr natural nenul <30000, $i \in \{1, 2, \dots, n\}$
- **m** – poate fi una din cifrele 2, 3, 4

Exemple

cadou.in	cadou.out
2	0
4	
32 81 100 19	
cadou.in	cadou.out
2	1
4	2 4
32 81 100 18	3 3
	5 1

Timp maxim de execuție/test: 1 secundă

cadouri-soluție

Soluția problemei este destul de simplă și este necesară utilizarea unor tablouri unidimensionale (vectori) pentru memorarea și prelucrarea datelor.

Se citesc din fișierul text cadou.in datele de intrare și se determină numerele prime ce pot apare în descompunerea în factori primi a datelor de intrare. Se descompune apoi fiecare număr în factori primi cumulând puterile acestor factori. Se verifică dacă fiecare factor prim apare la o putere multiplă de m și se scrie soluția în fișierul cadouri.out.

Pentru claritatea prezentării se utilizează subprograme.

Programul C++ este descris în continuare.

```

#include<iostream.h>
#include<fstream.h>
#include<math.h>
int x[6001], p[6001], y[6001], f[6001],m, n, dp, i, k,
j,sw;
void cit()
{int i;
ifstream f("cadou.in");
f>>m>>n;
for(i=1;i<=n;i++) f>>x[i];
f.close();
}
int prim(int k)
{int i=2,p=1;
while(p && i<=(int)sqrt(k))
{if (k % i==0 )
p=0;
i++;
}
return p;
}
}
void prime()
{int i, j;
p[1] = 2; p[2] = 3;
dp = 2; i = 5;
while (i<30000)
{
if (prim(i))
{ dp++;
p[dp] = i;
}
i = i+2;
}
}
int main()
{
cit();
prime();
for (i = 1 ;i<= dp; i++)
f[i] = 0;
}

```

```

for (i = 1; i <= n; i++)
{
    j = 1;
    while (x[i] != 1)
    {
        while (x[i] % p[j] == 0)
        {
            f[j]++;
            x[i] = x[i] / p[j];
        }
        j++;
    }
}
sw = 1;
for (i = 1; i <= dp; i++)
    if (f[i] % m != 0)
        {
            sw = 1;
            break;
        }
ofstream ff("cadou.out");
if (!sw) ff << 0 << endl;
else
{
    ff << 1 << endl;
    for (i = 1; i <= dp; i++)
        if (f[i] > 0)
            ff << p[i] << " " << f[i] / m << endl;
}
ff.close(); return 0;
}

```

PROBLEMA SORCOVA (clasa a IX-a)

prof. Gheorghe Manolache

Cei n elevi de la clasele unde predă profesorul StieTot vor să meargă cu sorcova. Fiecare elev trebuie să-și confecționeze o sorcovă din k ramuri cu flori de diverse tipuri nu neapărat distincte. Deoarece nu pot merge toți elevii deodată pentru a sorcovi profesorul, ei se decid să se împartă în grupe. Un elev se poate atașa unui grup numai dacă sorcova sa are cel puțin o floare de același tip cu cel puțin alt elev din grupul respectiv.

Cerință

Fiind dat un număr natural n reprezentând numărul elevilor și numărul natural k reprezentând numărul de flori ale unei sorcove, să se determine grupele care se formează.

Date de intrare

Fișierul de intrare **sorcova.in** conține pe prima linie, separate printr-un spațiu, numerele naturale n și k . Fiecare dintre următoarele n linii conține, pentru fiecare elev, câte k valori separate prin câte un spațiu reprezentând tipurile de flori culese pentru confecționarea sorcovei proprii

Date de ieșire

Fișierul de ieșire **sorcova.out** va conține pe fiecare linie câte o grupă formată din numerele de ordine ale elevilor separate prin câte un spațiu, în ordine crescătoare, ca în exemplu.

Restricții și precizări

- $1 \leq n \leq 150$
- $1 \leq k \leq 100$
- Tipul unei flori este un număr întreg din intervalul $[0, 100]$.
- Într-o grupă numerele de ordine ale elevilor trebuie date în ordine strict crescătoare.
- În fișierul de ieșire grupele vor fi afișate în ordinea crescătoare a numărului de ordine al primului elev din grupă.

Exemplu :

sorcova.in	sorcova.out	Explicații
5 4 1 2 3 4 5 6 9 6 1 1 1 1 2 4 4 3 7 7 7 7	1 3 4 2 5	Elevii 1 și 3 au folosit flori de tipul 1, iar elevii 1 și 4 au utilizat flori de tipurile 2,3 și 4, deci (1, 3, 4) se vor afla în aceeași grupă. Elevii 2 și 5 vor forma fiecare câte o grupă deoarece nu au ales flori de același tip cu nici unul dintre ceilalți elevi.

Timp de rulare/test: 1 secundă

sorcova- soluție

Soluția problemei utilizează tablouri.

Algoritmul problemei se poate descrie astfel:

1. se citesc n -numărul de elevi și k -numărul de flori din fiecare sorcovă.
2. se construiește matricea a definită astfel:
pe linia i sunt tipurile distincte de flori ale elevului cu numărul de ordine i .
 $a[i][0]$ =numărul de elemente de pe linia i ; acesta va deveni 0 dacă linia a fost reunită în alte linii.
3. se construiește vectorul viz care are n elemente. Inițial $viz[i]=i$ cu sensul că fiecare elev este în grupă singur; pe parcursul prelucrării, elevii care ajung în aceeași grupă vor avea aceeași valoare în vectorul viz (dacă elevul 6 ajunge în grupa în care este și elevul 4 atunci $viz[6]=viz[4]$);
4. se face apelul $irelj(i,j)$ prin care se verifică dacă i este în relație cu j : caută pe linia i și j un tip de floare comun elevilor i și j .
5. se realizează reuniunea mulțimilor de pe linia i și j în linia i folosind funcția reunește; dacă s-a făcut o astfel de reuniune,

decrementez valoarea i, astfel se rezolvă cazul în care de exemplu avem (i rel j), not (i rel k), (j rel k); decrementez i și astfel k va ajunge tot in grupa cu i (altfel k ar ajunge în altă grupă).

6. se afișează grupele prin selectarea din vectorul viz a pozițiilor care au aceeași valoare; pun pe 0 pozițiile afișate pentru a nu le afișa de mai multe ori.

Programul C++ este descris în continuare.

```
#include<iostream.h>
#include<stdio.h>
FILE *f=fopen("sorcova.in","r");
FILE *g=fopen("sorcova.out","w");
int n,k,a[150][150];
int irelj(int i,int j) //verifica daca i e in relatie cu j (au cel putin o floare in comun)
{int u,v;
for(u=1;u<=a[i][0];u++) //a[i][0] e numarul de elemente pe linia i
for(v=1;v<=a[j][0];v++)
if (a[i][u]==a[j][v])
return 1;
return 0;
}
int apartine(int val,int linie)
{int j,lg=a[linie][0]; //caut val in multimea de pe linia linie
for(j=1;j<=lg;j++)
if (val==a[linie][j])
return 1;
return 0;
}
void reuneste(int i,int j)
//reuneste in linia i linia j
{int u;
for(u=1;u<=a[j][0];u++)
if(!apartine(a[j][u],i))
{a[i][0]++;
a[i][ a[i][0] ]=a[j][u];
}
}
int main()
{int viz[150],i,j,val,ok;
fscanf(f,"%d %d",&n,&k);
for(i=1;i<=n;i++)
for(j=1;j<=k;j++)
{fscanf(f,"%d",&val);
if(!apartine(val,i))
{a[i][0]++;
//pe prima coloana am nr. de tipuri distincte de flori
a[i][ a[i][0] ]=val; //in multimea de pe linia i am tipurile distincte de flori al fetitei i
}
}
for(i=1;i<=n;i++)
viz[i]=i; //initial exista n grupe
for(i=1;i<=n;i++)
{ok=0;
if(a[i][0])
{
for(j=i+1;j<=n;j++)
if(irelj(i,j))
{
viz[j]=viz[i]; //j trebuie sa ajunga in grupa cu i
reuneste(i,j); //reunesc in linia i linia j
a[j][0]=0; //consider ca in multimea j am acum 0 elemente
ok=1;
}
}
if (ok) i--;
}
/*faptul ca am reunit in i cel putin o multime j implica sa continui cu aceeasi linie i, daca as lasa i sa se
incrementeze conform for-ului,ar gresi in sensul ca pt. i rel j si i nu e in rel cu k si j rel k ar pune j in grupa i dar k ar ajunge in
alta grupa*/
```

```

}
for(i=1;i<=n;i++)
  if(viz[i])
    {fprintf(g,"%d ",i);
    for(j=i+1;j<=n;j++)
      if(viz[i]==viz[j])
        {fprintf(g,"%d ",j);
        viz[j]=0; //ca sa nu mai fie prelucrat
        }
    fprintf(g,"\n");
  }
fclose(g); return 0;
}

```

PROBLEMA JOC (clasa a X-a)

prof. Gheorghe Manolache

Pentru copiii isteți, Mos Crăciun a inventat un joc prin care se propune realizarea unui șir de perle prin utilizarea unui șir de transformări. Avem la dispoziție 3 tipuri de perle normale (le vom nota cu 1, 2 și 3) și 3 tipuri de perle magice (le vom nota cu A, B și C). Perlele magice sunt deosebite prin faptul că se pot transforma în alte perle (una sau mai multe, normale sau magice).

Perla magică de tipul A se poate transforma în orice perlă normală (una singură).

Perla magică de tipul B se poate transforma într-o perlă normală de tipul 2 și una magică de tipul B, sau într-o perlă normală de tipul 1, una magică de tipul A, una normală de tipul 3, una magică de tipul A și una magică de tipul C.

Perla magică de tipul C se poate transforma într-o perlă normală de tipul 2 sau într-o perlă normală de tipul 3, una magică de tipul B și una magică de tipul C sau într-o perlă normală de tipul 1, una normală de tipul 2 și una magică de tipul A. Mai sugestiv, putem scrie relațiile următoare:

A -> 1 | 2 | 3
 B -> 2B | 1A3AC
 C -> 2 | 3BC | 12A

Moșul lasă copiii să aleagă inițial o perlă magică (una singură), iar apoi folosind numai transformările de mai sus trebuie să obținem un anumit șir de perle normale. Când o perlă magică se transformă, perlele din stânga și din dreapta ei rămân la fel (și în aceeași ordine). De asemenea ordinea perlelor rezultate din transformare este chiar cea prezentată mai sus. Dacă moșul ne cere să facem șirul de perle 21132123, putem alege o perlă magică de tipul B și următorul șir de transformări: B -> 2B -> 21A3AC -> 21A3A12A -> 21132123. Moșul nu cere decât să spunem dacă se poate sau nu obține șirul respectiv de perle.

Cerință

Să se determine pentru fiecare șir de intrare dacă se poate obține prin transformările de mai sus sau nu (alegând orice primă perlă magică, la fiecare șir).

Date de intrare

Fișierul de intrare **joc.in** are următoarea structură:

- pe prima linie numărul **N**, reprezentând numărul de șiruri din fișierul de intrare
- urmează **N** linii; a **i**-a linie dintre cele **N** descrie șirul **i**, printr-o succesiune de numere naturale despărțite de câte un spațiu. Primul număr reprezintă lungimea șirului L_i , iar următoarele L_i numere sunt tipurile de perle normale, în ordine, de la stânga la dreapta.

Date de ieșire

Fișierul **joc.out** va conține **N** linii. Pe linia **i** se va scrie un singur număr **1** sau **0** (**1** dacă se poate obține șirul respectiv (al **i**-lea) și **0** dacă nu se poate).

Restricții

- $0 < N < 11$
- $0 < L_i < 10001$, pentru oricare **i**

Exemplu

joc.in	joc.out
3	1
8 2 1 1 3 2 1 2 3	0
2 2 2	1
1 3	

Timp maxim de execuție/fișier test: 1 secundă.

Rezolvarea problemei se bazează pe lucrul cu stiva.

Programul C++ descris în continuare.

```
#include <stdio.h>
#define NMAX 10000
```

```
int st[2][NMAX+10]; /* 1, 2, 3 e evident si 4, 5, 6 e A, B, C */
```


Avem relațiile de recurență:

$$Ns(k)=Ns(k-1)*9+Nd(k-1)*9;$$

$$Nd(k)=Ns(k-1)$$

Prima relație spune că din fiecare număr de $k-1$ cifre de primul tip putem obține 9 numere de același tip (adaugând oricare din cele 9 cifre diferite de ultima cifră), la fel putem obține și din fiecare număr de al doilea tip încă 9 numere.

A doua relație spune că putem obține din fiecare număr de primul tip cu $k-1$ cifre un număr de al doilea tip, adăugând încă o cifră egală cu ultima cifră.

Rezultatul cerut se obține adunând $Ns(n)+Nd(n)$, evident e nevoie de operații cu numere mari, cu cel mult 501 cifre (numărul cerut e mai mic decât numărul tuturor numerelor de n cifre).

Să mai observăm că nu e nevoie de memorarea într-un vector a valorilor $Ns(i)$ și $Nd(i)$, deoarece nu avem nevoie decât de valorile precedente.

Algoritmul PSEUDOCOD aferent este următorul:

```

citeste n
ns=9
nd=0 (pentru o cifră avem 9 numere de tip 1 și 0 de tip 2)
pentru i=2,n execută
    aux=ns
    ns=(ns+nd)*9
    nd=aux
sfirsit pentru
scrie ns+nd
sfirsit algoritmul

```

Programul C++ (aferent problemei **numere**)

```

#include <stdio.h>
int ns[501],nd[501],aux[501],x[501],i,j,n;
FILE *f;
void sum(int a[],int b[],int c[])
{int i,r,t=0;
for(i=0;i<=500;i++) {r=a[i]+b[i]+t; c[i]=r%10; t=r/10;}
}
void prod(int a[],int k,int b[])
{int i,r,t=0;
for(i=0;i<=500;i++){r=a[i]*k+t; b[i]=r%10; t=r/10;}
}
void main()
{f=fopen("numere.in","rt"); fscanf(f,"%d",&n); fclose(f);
for(i=0;i<=500;i++) ns[i]=nd[i]=aux[i]=x[i]=0;
ns[0]=9;
for(i=2;i<=n;i++)
{for(j=0;j<=500;j++) aux[j]=ns[j];
sum(ns,nd,x); prod(x,9,ns);
for(j=0;j<=500;j++)nd[j]=aux[j];
}
sum(ns,nd,aux);
i=500;
while(!aux[i]) i--;
f=fopen("numere.out","wt");
while(i>=0) {fprintf(f,"%d",aux[i]); i--;}
fclose(f);
}

```

PROBLEMA GARAJ (clasele XI-XII)

prof. Dumitru Ene

Costel este administratorul unui garaj dreptunghiular de dimensiuni N (metri) \times 4 (metri), format din blocuri de beton pătrate, cu latura de un metru. Garajul are o singură ieșire pe una din laturi, în rest este înconjurat de ziduri pe care autovehiculele nu le pot depăși. În garaj sunt parcate autovehicule, notate cu A,B,C,

Autovehiculul A este un pătrat cu latura de 1m, iar celelalte (B,C, ...) sunt dreptunghiuri de lungime 2m și lățime 1m. Autovehiculul A se poate deplasa în cele 4 direcții (N- nord, S-sud, E-est, V-vest), pe când celelalte se deplasează doar în direcția lungimii lor.

Astăzi Costel este într-o situație delicată: șeful i-a cerut să scoată autovehiculul A din garaj. Costel a încercat timp de o oră să realizeze acest lucru, dar nu a reușit și s-a gândit să apeleze la ajutorul vostru. El a făcut o hartă a garajului: a reprezentat cu ' . ' un bloc de beton, cu ' # ' zidurile garajului și cu litere A, B, C, ... autovehiculele.

Cerință

Scrieți un program care să scoată autovehiculul A din garaj în cât mai puține mișcări. Se consideră o mișcare mutarea unui autovehicul, într-o anumită direcție, cu un număr întreg de metri.

Date de Intrare

Pe prima linie a fișierului `garaj.in` se află numărul N reprezentând lungimea garajului.

Pe următoarele $N+2$ linii avem redată harta garajului, fiecare linie conținând câte 6 caractere.

Date de Ieșire

Pe prima linie a fișierului `garaj.out` se va afla numărul de mișcări P prin care a fost rezolvată problema. Pe următoarele P linii se vor afla 3 caractere ce reprezintă o mutare. Primul reprezintă litera asociată autovehiculului ce va fi mutat, al doilea direcția în care va fi mutat (N- nord, S-sud, E-est, V-vest), iar al treilea numărul de blocuri cu care va fi mutat. În locul mișcării prin care autovehiculul A părăsește garajul se va afișa 'Exit'.

Restricții și precizări

- În fișierul de intrare se va găsi un caracter ' . ' (ieșirea) în locul unui caracter ' # ' de pe frontiera garajului;
- Sunt maxim 6 mașini în parcare.
- Dacă un autovehicul are o anumită literă a alfabetului atunci există exact un autovehicul atribuit fiecărei litere mai mici decât litera respectivă.
- Este imposibil ca alt autovehicul să părăsească garajul înaintea autovehiculului A;
- Nu există două autovehicule cu aceeași literă, iar un autovehicul nu poate trece peste altul în deplasarea lui;
- $4 \leq N \leq 8$;
- Se garantează existența unei soluții pentru fiecare test.

Exemplu:

garaj.in	garaj.out
6	4
##.###	CS1
#....#	BE1
#D...#	AV1
#DBEC#	Exit
#...C#	
#..A.#	
#....#	
#####	

Timp maxim de execuție/test: 1 secundă

garaj-soluție

Problema este exponențială în dimensiunea matricii, dar polinomială în numărul total de posibilități de poziționare al mașinilor. Astfel, vom folosi un algoritm de tip BFS care garantează că se ajunge la soluție într-un număr minim de mișcări. Plecăm de la matricea inițială, și expandăm pe rând toate stările posibile, mișcând din starea curentă câte o mașină până când nu mai există nici o variantă nouă de poziționare a mașinilor sau până când am scos mașina A din parcare. Stările problemei le putem codifica într-un întreg de 32 de biți. Singurele variabile sunt pozițiile mașinilor. După ce eliminăm zidurile înconjurătoare, coordonatele nu sunt mai mari de 7 (3 biți) pentru linii și 3 (2 biți) pentru coloane, deci pentru poziția unei mașini vom folosi 5 biți. Concatenăm pozițiile mașinilor și, cum sunt maxim 6 mașini, codificarea nu va avea mai mult de 30 de biți.

Pentru a memora stările explorate vom folosi o tabelă de dispersie. De precizat și că numărul total de posibilități pornind de la starea inițială este destul de redus, deci problema va rula în timpul cerut.

Programul C++ (afereț problemei **garaj**)

```
#include<fstream.h>
# define M 3381
long T[M];
ifstream f("garaj.in"); ofstream g("garaj.out");
char a[12][8],sA[5]="NESV",cod[7]="ABCDEFGH";
struct art {int l,c;} v[6];
struct coada {long st; unsigned int t; char cod,dir; int nrb;}
q[M];
struct stiva {long st; char cod,dir; int nrb;} s[200];
int N,lf,cf,nm=0,w=1,sens[6]={0,0,0,0,0,0};
int dl[4]={-1,0,1,0},dc[4]={0,1,0,-1};
long stare,p,u;
int h(long k, int i)
{return 1+(k%M + i * (1+k%(M-1)))%M;}
int insT(long k)
{int j,i=0;
do {j=h(k,i);
if(T[j]==0) {T[j]=k; return j;}
else i++;
} while(i<M);
cout<<"Depasire tabela de dispersie\n"; return 0;
}
int citT(long k)
{int j,i=0;
do {j=h(k,i);
if(T[j]==k) return j;
i++;
} while(T[j]!=0 && i<M);
return 0;
}
long codificare(art v[])
```

```
{long y=0;
for(int i=0;i<=nm;i++) y=y*32+(v[i].l-2)*4+(v[i].c-2);
return y;
}
void decodificare(long y)
{unsigned int x; int i,j;
for(i=2;i<N+2;i++) for(j=2;j<6;j++) a[i][j]='.';
for (i=nm; i>0;i--)
{x=y%32; y /=32;
v[i].c=x%4+2; v[i].l=x/4+2;
a[v[i].l][v[i].c]='A'+i;
if(sens[i]) a[v[i].l+1][v[i].c]='A'+i;
else a[v[i].l][v[i].c+1]='A'+i;
}
x=y%32; v[0].c=x%4+2; v[0].l=x/4+2;
a[v[0].l][v[0].c]='A';
a[lf][cf]='.';
}
void cit()
{int i,j,x,nr[6]={0,0,0,0,0,0};
f>>N; f.get();
for(i=1;i<=N+2;i++) {for(j=1;j<=6;j++)
a[i][j]=f.get(); f.get();}
for(j=1;j<=6;j++)
{a[0][j]='*'; if(a[1][j]=='.') {lf=1; cf=j;}
a[N+3][j]='*'; if(a[N+2][j]=='.') {lf=N+2; cf=j;}
}
for(i=0;i<=N+3;i++)
{a[i][0]='*'; if(a[i][1]=='.') {lf=i; cf=1;}
a[i][7]='*'; if(a[i][6]=='.') {lf=i; cf=6;}
}
for (i=2;i<N+2;i++)
```

```

for(j=2;j<6;j++)
    {x=a[i][j]-'A'; if(x>nm) nm=x;
    if(x==0) {v[0].l=i; v[0].c=j;}
    if(x>0)
        if(!nr[x]) {v[x].l=i; v[x].c=j; nr[x]++;}
        else if(v[x].c==j) sens[x]=1;
    };
stare=codificare(v);
}
void rezolv()
{int ll,cc,k,ok,nb,i; art vv[6]; long st;
memset(T,0,M);
q[1].st=stare; q[1].t=0; p=u=1;
insT(stare);
while(w)
{decodificare(q[p].st);
for(i=0; i<4 && w; i++) //explozie masina A
{ll=v[0].l; cc=v[0].c; ok=1; nb=0;
for(k=0; k<=nm; k++) vv[k]=v[k];
do {ll+=dl[i]; cc+=dc[i]; nb++;
if(a[ll][cc]=='.') {if(ll==lf && cc==cf)
{w=0; u++;
q[u].cod='A'; q[u].t=p;
q[u].dir=sA[i]; q[u].nrb=nb;
}
else
{vv[0].l=ll; vv[0].c=cc;
st=codificare(vv);
if(citT(st)==0)
{u++; q[u].st=st;
q[u].t=p; q[u].cod='A';
q[u].dir=sA[i]; q[u].nrb=nb;
insT(st);
}
}
}
else ok=0;
}
while(ok && w);
}
a[lf][cf]='#';
for(i=1; i<=nm && w; i++) //explozie masini B,C,...
{if(sens[i]) // vertical
{ll=v[i].l; cc=v[i].c; ok=1; nb=0;
for(k=0; k<=nm; k++) vv[k]=v[k];
do {ll--; nb++;
if(a[ll][cc]=='.')
{vv[i].l=ll; vv[i].c=cc;
st=codificare(vv);
if(citT(st)==0)
{u++; q[u].st=st; q[u].cod=cod[i];
q[u].t=p; q[u].dir='N'; q[u].nrb=nb;
insT(st);
}
}
else ok=0;
}
while(ok);
ll=v[i].l; cc=v[i].c; ok=1; nb=0;
for(k=0; k<=nm; k++) vv[k]=v[k];
do {ll++; nb++;
if(a[ll+1][cc]=='.')
{vv[i].l=ll; vv[i].c=cc;
st=codificare(vv);
if(citT(st)==0)
{u++; q[u].st=st; q[u].cod=cod[i];
q[u].t=p; q[u].dir='S'; q[u].nrb=nb;
insT(st);
}
}
else ok=0;
}
while(ok);
ll=v[i].l; cc=v[i].c; ok=1; nb=0;
for(k=0; k<=nm; k++) vv[k]=v[k];
do {cc--; nb++;
if(a[ll][cc]=='.')
{vv[i].l=ll; vv[i].c=cc;
st=codificare(vv);
if(citT(st)==0)
{u++; q[u].st=st;
q[u].cod=cod[i]; q[u].t=p;
q[u].dir='V'; q[u].nrb=nb;
insT(st);
}
}
else ok=0;
}
while(ok);
ll=v[i].l;
cc=v[i].c;
ok=1;
nb=0;
for(k=0; k<=nm; k++) vv[k]=v[k];
do {cc++; nb++;
if(a[ll][cc+1]=='.')
{vv[i].l=ll; vv[i].c=cc;
st=codificare(vv);
if(citT(st)==0)
{u++;
q[u].st=st;
q[u].cod=cod[i];
q[u].t=p;
q[u].dir='E';
q[u].nrb=nb;
insT(st);
}
}
else ok=0;
}
while(ok);
}
p++;
}
}

void afisare()
int nr=0,r=u;
do {nr++;
s[nr].st=q[r].st; s[nr].cod=q[r].cod;
s[nr].dir=q[r].dir; s[nr].nrb=q[r].nrb;
r=q[r].t;
}
while(r);
nr--;
g<<nr<<'\n';
while(nr>1)

```

```

{decodificare(s[nr].st);
g<<s[nr].cod<<s[nr].dir<<s[nr].nrb<<'\n';
nr--;
}
g<<"Exit\n"; g.close();
}

int main()
{cit();
rezolv();
afisare();
return 0;}

```

PROBLEMA HIPIC (clasele XI-XII)

prof. Dumitru Ene

La tradiționalul concurs hipic ce se organizează anual la Piatra Neamț se montează N obstacole numerotate de la 1 la N . Pentru a parcurge traseul concursului, călăreții pot începe cu orice obstacol, trebuie să treacă peste toate obstacolele, iar de la un obstacol la altul pot circula într-un singur sens, stabilit înaintea concursului.

Cerință

Să se precizeze un traseu pe care îl poate urma un călăreț.

Date de intrare

Fișierul de intrare `hipic.in` conține pe prima linie N (numărul de obstacole), iar pe fiecare dintre următoarele $N*(N-1)/2$ linii o pereche de numere naturale $x\ y$ cu semnificația "obstacolul x poate fi sărit imediat înaintea de a sării obstacolul y ".

Date de ieșire

Fișierul de ieșire `hipic.out` conține pe o singură linie traseu pe care îl poate urma un călăreț.

$z_1\ z_2\ \dots\ z_N$ (z_1 este primul obstacol al traseului)

Restricții

- $2 \leq N \leq 150$
- Valorile scrise pe aceeași linie în fișierele de intrare și ieșire sunt separate prin spații.
- Dacă există mai multe soluții, se va determina una singură.

Exemplu

hipic.in	hipic.out
4	4 3 1 2
1 2	
1 4	
3 1	
2 4	
3 2	
4 3	

Timp maxim de execuție/test: 1 secunda

hipic-soluție

Avem o problemă clasică de graf turneu prezentată și în manualul de clasa a XI-a.

Programul C++ (afereț problemei **hipic**)

```

#include<fstream.h>
#define NMAX 151
ifstream f("hipic.in"); ofstream g("hipic.out");
int n,i,j,x,y,k,m=2,ok;
int a[NMAX][NMAX],D[NMAX];

int main()
{f>>n;
for(i=1; i<=n; i++)
for(j=1; j<=n; j++)
a[i][j]=0;
for(i=1; i<=n*(n-1); i++)
{f>>x>>y; a[x][y]=1;}
f.close();
if(a[1][2]) {D[1]=1; D[2]=2;}
else {D[1]=2;
D[2]=1;}

for(i=3; i<=n; i++)
{if(a[i][D[1]]) k=1;
else {j=1; ok=0;
while(j<m && !ok)
if(a[D[j]][i] && a[i][D[j+1]])
ok=1;
else j++;
k=j+1;
}
for(j=m+1; j>k; j--) D[j]=D[j-1];
D[k]=i; m++;
}
for(i=1; i<=n; i++)
g<<D[i]<<' ';
g<<'\n';
g.close();
return 0;}

```

PROFESOR – ELEV – COMPUTER, EXPRESIA GENERALĂ A MILENIULUI III

*Rotariu Mihai, Zaiț Mădălina
(clasa a X-a F)*

Sunt aproximativ 20 ani de când computerul a pătruns în învățământ. Dacă la început el era gândit ca un instrument de lucru pentru ceea ce erau numite atunci *laboratoare de informatică*, unde aveau acces elevii care se pregăteau în acest domeniu, ultimii ani au adus o adevărată revoluție conceptuală în educație, computerul devenind un mediu pentru învățare care reușește să acopere toate disciplinele din programă.

Impresionanta dezvoltare a tehnologiilor multimedia a facilitat și ea apariția unui domeniu nou: *softul educațional*. Aceasta noțiune denotă un foarte interesant domeniu hibrid, între programarea informatică, psihopedagogia și diverse materii de curricula, domeniu, care este pe cale să devină unul cu drepturi depline și viitor sigur în oferta educațională a acestui început de mileniu.

Ținând cont de aspectele precizate mai sus, alături de colegul meu am realizat un soft educațional pentru disciplina INFORMATICĂ, având ca tema principală metoda de programare “*Backtracking*”.

Acest proiect ne-a adus locul al III-lea la concursul “**Pro Soft@NT**” – **secțiunea Soft Utilitar și Pagini Web**, concurs organizat de zilele Colegiului Național de Informatică.

Programul nostru are rolul de a contribui la învățarea și asimilarea unor informații legate de această metoda de programare, urmărind obiective precise care pot aduce rezultate cuantificabile.

De asemenea sunt convinsă că sunteți cu toții de acord ca utilizarea acestor softuri tematice este o metoda comodă și foarte eficientă atât din perspectiva profesorului cât și a elevilor. Nivelul de cunoștințe asimilate în urma folosirii unui astfel de program este unul mult mai ridicat, comparativ cu al unor lecții obținute deoarece unele reprezentări pot fi reproduse sau simulate doar prin intermediul calculatorului care oferă metode și tehnici privind grafica, animația, sunetul.

În ceea ce privește modul de realizare al proiectului nostru... veți putea observa că fiecare pagină conține în partea stângă câteva noțiuni de teorie... iar în partea dreaptă aplicații ale informațiilor regăsite în prima jumătate. Acest mod de prezentare dă ocazia tuturor celor care folosesc acest soft să își verifice, într-un mod cât mai interactiv, cunoștințele însușite în urma parcurgerii noțiunilor de teorie. În acest scop în partea de jos se găsesc două butoane: butonul “undo” și butonul de validare. Cu ajutorul acestora utilizatorul poate să verifice dacă răspunsurile lui au fost corecte sau incorecte iar în cazul în care dorește ca pagina să revină la forma inițială poate apăsa butonul “undo”.

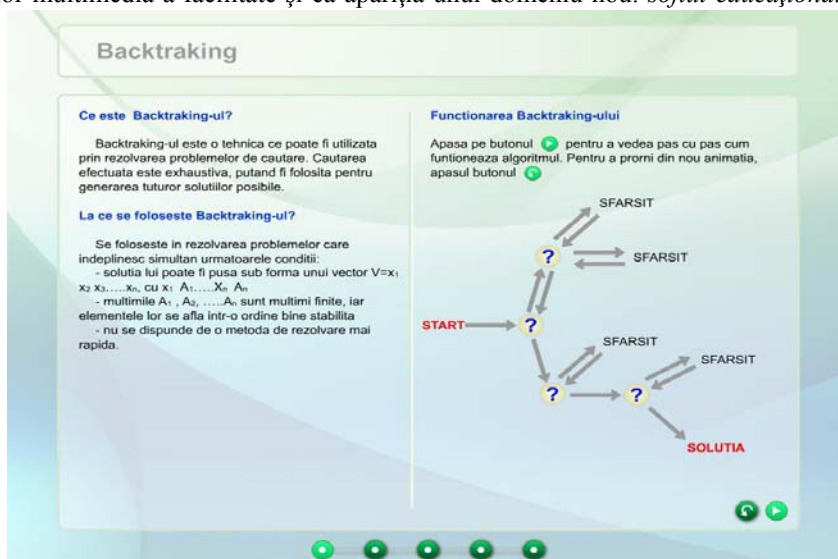
După parcurgerea întregului program de informare... la sfârșit programul nostru cuprinde un test care ne poate ajuta să stabilim feedback-ul obținut. Răspunsurile corecte ale tuturor întrebărilor introduse de către noi, ne garantează eficiența pe care a avut-o softul nostru.

Ce aș mai putea adăuga... referitor la programul nostru sunt avantajele pe care acesta le are:

- o interfață primitoare;
- nuanțe cromatice plăcute care permit o bună vizualizare a tuturor informațiilor;
- butoanele, care sunt așezate în aceeași poziție tocmai cu un scop ergonomic (oferind utilizatorului o eficiență și o ușoară parcurgere a informațiilor care se va desfășura în condiții optime);

Astfel privind în ansamblu softul conceput de noi... considerăm ca cel mai mare avantaj al programului este ca profesorul are la dispoziție o platformă de învățare. El își poate alege să prezinte părți de lecție pe teme din programă dar poate, la rândul lui, să își creeze diverse fragmente în funcție de feedbackul primit de la un anumit grup de elevi sau în funcție de strategia pe care și-o propune...

În încheiere dorim să subliniem încă o dată ca folosindu-ne de acest soft interactiv putem învăța și vedea practic care sunt utilizările metodei de programare, backtracking, precum și caracteristicile acestei metode.



TERENURI

Ștefan Mihăilă (clasa a XI-a F)

Fănel are de cumpărat un teren, pe care să își planteze flori. El are de ales dintre N terenuri de formă pătratică. Fiecare dintre acestea este caracterizat printr-un număr real, ce reprezintă aria terenului și prin poligoane convexe care sunt considerate suprafețe pe care nu se poate planta. Aria reală a terenului este dată de diferența dintre aria totală și ariile poligoanelor convexe.

Cerință

Ajută-l pe Fănel să găsească terenul ce are aria reală maximă.

Date de intrare

terenuri.in	Explicație
N $S_1 S_2 S_3 \dots S_N$ $P_1 P_2 \dots P_N$ $A_1 X_{11} Y_{11} X_{12} Y_{12} \dots X_{1,A[1]} Y_{1,A[1]}$...	Numărul de terenuri Ariile celor N terenuri Numărul de poligoane pentru fiecare din cele N terenuri Un număr (A_i) semnificând din câte puncte este format primul poligon, urmat de $2 \cdot A_i$ numere reale reprezentând coordonatele acelor puncte Vor fi $P_1 + P_2 + \dots + P_N$ de astfel de linii.

Date de ieșire

Prima linie a fișierului terenuri.out va conține indicele (începând de la 1) al terenului cu arie reală maximă. Cea de-a doua linie va conține aria reală a terenului, cu 2 zecimale exacte.

Exemplu

terenuri.in	terenuri.out
4 25 36 49 100 1 2 2 0 4 1 4 4 4 1 1 4 1 3 2.5 1 1 5 4 5 4 4 0 6 4 6 0 4 4 3 0 2 0 7 3 7 3 0 0 7 0 3.5 7	3 100.00

Restricții

- $1 \leq N \leq 20$
- $1 \leq S_i \leq 100000$
- Un teren conține cel mult 50 poligoane convexe ($0 \leq P_i \leq 50$)
- Poligoanele convexe date pentru un teren nu se vor intersecta
- Un poligon este dat prin cel mult 20 puncte ($3 \leq A_i \leq 20$)
- Coordonatele punctelor unui poligon vor fi date în ordine aleatoare
- Oricare 3 vârfuri ale unui poligon sunt necolineare.
- $0 \leq X_{ij}, Y_{ij} \leq 2000$

Soluție

Pentru aria reală maximă, vom calcula aria reală a fiecărui teren. Vom afla aria fiecărui poligon, și o vom scădea din aria totală a terenului din care face parte.

Calcularea ariei unui poligon necesită sortarea în sens trigonometric, împărțirea acestuia în triunghiuri, și însumarea ariilor triunghiurilor obținute.

```

#include <iostream.h>
#include <fstream.h>
#include <algorithm.h>
#include <iomanip.h>

using namespace std;

const unsigned int MAXTERENURI = 20;
const unsigned int MAXPOLY = 50;
const unsigned int MAXPUNCTE = 21;

struct Punct { double x,y; };
struct Poly
    { unsigned int n; Punct p[MAXPUNCTE]; };

Punct p0;

/*
punctul de referinta
(in functie de care se calculeaza pantele)
|x0 y0 1|
|x1 y1 1| = (x1-x0)*(y2-y0) - (y1-y0)*(x2-x0)
|x2 y2 1|
*/

bool trig (const Punct &p1, const Punct &p2)
{
return (p1.x-p0.x)*(p2.y-p0.y)-(p1.y-p0.y)*(p2.x-p0.x)>0;
}

int main ()
{
ifstream f_in("terenuri.in");
ofstream f_out("terenuri.out");
unsigned int N, nrpoly[MAXTERENURI], i, j;
double S[MAXTERENURI], maxS;
unsigned int maxS_teren;
Poly poly[MAXPOLY*MAXTERENURI];

f_in >> N;
for (i=0; i<N; ++i)
    f_in >> S[i];

unsigned int total_poly = 0;
for (i=0; i<N; ++i)
    f_in >> nrpoly[i], total_poly += nrpoly[i];

unsigned int teren_curent = 0;
for (i=0; i<total_poly; ++i)
{
// verificam din care teren face parte poligonul

    if (nrpoly[teren_curent])
        --nrpoly[teren_curent];

    else do
        { ++teren_curent; }
        while (!nrpoly[teren_curent]);

//citim poligonul
    unsigned int nr_puncte;
    f_in >> nr_puncte;
    for (j=0; j<nr_puncte; ++j)

// coordonatele x si y ale punctului j din poligonul i

        f_in >> poly[i].p[j].x >> poly[i].p[j].y;
        if (S[teren_curent] > 0)
            {

/*sortam punctele in sens trigonometric sau al acelor de
ceas
alegem un punct din interiorul poligonului ca origine pen-
tru sortare
fie el mijlocul segmentului dat de primele 2 puncte ale
poligonului
*/

                p0.x = (poly[i].p[0].x + poly[i].p[1].x)/2;
                p0.y = (poly[i].p[0].y + poly[i].p[1].y)/2;
                sort(poly[i].p, poly[i].p+nr_puncte, trig)

//facem ca ultimul punct sa fie acelasi cu primul

                poly[i].p[nr_puncte].x = poly[i].p[0].x;
                poly[i].p[nr_puncte].y = poly[i].p[0].y;
                ++nr_puncte;

/*
calculam aria poligonului
formula se obtine impartind poligonul in triunghiuri
si calcularea ariei fiecarui triunghi (cu determinanti)
*/

                double arie=0.;
                for (j=0; j<nr_puncte-1; ++j)

//arie += x(j)*y(j+1) - x(j+1)*y(j)

                    arie += (poly[i].p[j].x*poly[i].p[j+1].y-
                        poly[i].p[j+1].x*poly[i].p[j].y);
                    if (arie<0) arie *= -1;
                    arie /= 2.;

// o scadem din aria terenului din care face parte

                    S[teren_curent] -= arie;
                }
            }

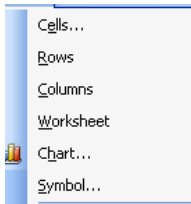
        f_in.close();
        maxS = S[0];
        maxS_teren = 0;
        for (i=1; i<N; ++i)
            if (S[i] > maxS)
                maxS = S[i], maxS_teren = i;
        f_out << maxS_teren+1 << '\n' ;
        f_out << setprecision(2) << fixed << maxS;
        f_out.close();
        return ();
    }
}

```

MODEL DE SUBIECT

clasa a X-a

prof. Mîndru Mariana

1. **Ce reprezintă AQ12 în Microsoft Excel?:**
 - a. o informație de tip text
 - b. un domeniu format din 12 celule
 - c. adresa unei celule
2. **Nu este caracteristic unei funcții Excel faptul că:**
 - a. argumentele sunt obligatorii
 - b. începe întotdeauna cu semnul "="
 - c. numărul argumentelor diferă de la o funcție la alta
 - d. argumentele funcției pot fi și valori logice
3. **Pentru activarea protecției pentru o foaie de calcul se selectează:**
 - a. tasta F7
 - b. din meniul Tools comanda Protect document
 - c. din meniul Tools comanda Protection
 - d. nici una dintre comenzile de mai sus
4. **Ce efect are modificarea datelor care intră într-o formulă?**
 - a. nu are nici un efect
 - b. se modifică și rezultatul
 - c. nu pot fi modificate datele care intră într-o formulă
 - d. produce eroare
5. **Care dintre următoarele funcții Excel returnează data curentă ?**
 - a. TODAY b. DATE c. MOD d. NOW
6. **Într-o diagramă pot fi introduse date editate ulterior creării ei?**
 - a. Da
 - b. Nu
7. **Din ce meniu fac parte comenzile din caseta alăturată?**
 - a. Format
 - b. Insert
 - c. Tools
 - d. Data
8. **În Microsoft Excel un număr precedat de semnul apostrof este tratat drept:**
 - a. text b. număr c. parte dintr-o formulă
9. **Următoarea formulă este greșită: =sheet1!B5*sheet3!B2**
 - a. da
 - b. nu
10. **În Microsoft Excel ce efect are următoarea combinație de taste: CTRL>+<:>**
 - a. nici un efect
 - b. introduce ora curentă
 - c. introduce data curentă
11. **Ce efect produce activarea opțiunii Match case?**
 - a. operația de căutare este dependentă de introducerea datelor cu majuscule sau cu minuscule
 - b. chiar dacă editarea celulei se face cu litere mici, la validare literele se vor transforma în majuscule
 - c. schimbă literele din majuscule în minuscule și invers
12. **Pentru un registru nou, numărul de foi de calcul este o valoare prestabilită la:**
 - a. 16
 - b. 4
 - c. poate fi modificată această valoare
13. **Sortarea datelor în Excel se poate face după:**
 - a. cel mult două criterii de sortare
 - b. cel mult trei criterii de sortare
 - c. oricâte criterii de sortare
14. **Adresa unei celule este formată din:**
 - a. numărul corespunzător coloanei și litera corespunzătoare rândului
 - b. litera corespunzătoare coloanei și numărul corespunzător rândului
 - c. numărul corespunzător rândului și litera corespunzătoare coloanei
15. **#N/A! este o eroare care are semnificația:**
 - a. nici o valoare disponibilă
 - b. eroare într-un număr
 - c. argumentul din formulă conține o greșeală
 - d. nici unul dintre răspunsurile de mai sus

16. Care dintre funcțiile următoare ce adună numere din domeniul D1 până la D4 este eronată?
- SUM(D1,D2,D3,D4)
 - SUM(D1;D4)
 - SUM(D1:D4)
 - Nici una
17. Ce rezultat se obține în urma aplicării următoarei formule: $36^{1/2}$
- 6
 - 13
 - 18
18. Care este rezultatul funcției =IF(2;1;2)
- 1
 - 2
 - #VALUE
19. Care este modul de vizualizare din care se poate face ștergerea unui diapozitiv în Microsoft PowerPoint?
- Master View
 - Slide Sorter View
 - Slide Show
20. La comutarea dintr-un mod de vizualizare în altul care diapozitiv rămâne activ?
- diapozitivul curent
 - primul diapozitiv
 - ultimul diapozitiv
21. Ce tip de informație poate să conțină un slide?
- text, grafică,
 - sunet, clipuri video
 - un registru Excel
 - toate răspunsurile de mai sus
22. Care este valoarea de adevăr a următoarei afirmații?
Odată creată prezentarea nu mai este posibilă reordonarea slide-urilor.
- adevărat
 - fals
23. Pentru proiectarea antetului și subsolului unei prezentări electronice se folosește:
- View → Header and Footer
 - Insert → Autotext → Header and Footer
 - Nu există în Microsoft PowerPoint această posibilitate
24. Ce este Pen color?
- un instrument cu care se poate desena liber pe diapozitive la editare
 - un instrument care permite sublinierea textelor importante în momentul expunerii
 - un instrument de selecție a unei zone de pe diapozitiv
25. Tabelele pot fi introduse pe un slide selectând:
- Insert → Table
 - Table → Insert → Table
 - nici o comandă de mai sus
26. Pentru a adăuga elemente în mișcare în prezentare puteți utiliza:
- Insert → Picture
 - Animation → Schemes
 - Slide transitions
 - Toate de mai sus
27. Pentru ca o anumită imagine să apară ca background în fiecare slide, care este cea mai eficientă modalitate:
- Inserați imaginea în fiecare pagină folosind Insert-Picture-From File
 - Inserați imaginea doar în Master slide
 - Folosiți Format-Background
28. Avansul la următoarea pagină în timpul prezentării se realizează:
- La apăsarea unei taste
 - După un timp prestabilit
 - La click
 - Toate variantele de mai sus
29. Pentru a tipări miniaturi ale slide-urilor, care este comanda potrivită?
- Print → Handouts
 - Print → Notes Pages
 - Print → slides
30. Care dintre următoarele tipuri de fișiere pornește direct în modul slide show?
- *.ppt
 - *.pps
 - *.pot

MODEL DE SUBIECT

*prof. Elena Andone
prof. Vasile Diaconu*

În sprijinul elevilor de clasa a XII-a prezentăm un model rezolvare a unei variante de subiect de bacalaureat.

Subiectul I (30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Știind că inițial variabilele întregi x , y și z au valorile $x=1$, $y=2$ respectiv $z=3$, în ce ordine trebuie scrise atribuirile următoare astfel încât, în final, expresia $x+y+z$ să aibă valoarea maximă? (4p.)

I) $x=x+y-z$; II) $y=x-y+z$; III) $z=z-x+y$;

- a. III II I b. I II III c. III I II d. II I III

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

- a) Scrieți care este valoarea afișată în urma executării algoritmului dacă se citește numărul 199. (6p.)
- b) Scrieți cel mai mic și cel mai mare număr, fiecare având exact 3 cifre, care pot fi citite astfel încât, în ambele cazuri, să se afișeze valoarea 7. (6p.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să utilizeze o singură structură repetitivă. (4p.)

citește a
(număr natural, $a < 10^9$)

```

repetă
| b ← 0
| cât timp a ≠ 0 execută
| | b ← b + a % 10
| | a ← [a / 10]
| ■
| a ← b
până când a < 10

```

scrie b

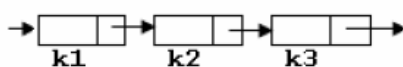
Subiectul II (30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera care corespunde răspunsului corect.

1. Un graf neorientat este complet dacă oricare două noduri distincte ale sale sunt adiacente. Care este numărul minim de muchii care trebuie eliminate dintr-un graf neorientat complet cu 100 de noduri astfel încât graful parțial obținut să fie eulerian? (4p.)

- a. 4851 b. 0 c. 100 d. 50

2. Fiecare element al unei liste simplu înlănțuite memorează în câmpul urm adresa următorului element din listă, iar variabilele k_1 , k_2 și k_3 , rețin adresele a 3 elemente succesive în listă, ca în figură. Cu ce instrucțiune se pot înlocui punctele de suspensie din secvența de mai jos astfel încât aceasta să determine interschimbarea corectă a pozițiilor în listă a elementelor de la adresele k_2 și k_3 ? (4p.)



$k_1 \rightarrow \text{urm} = k_3$; ... $k_3 \rightarrow \text{urm} = k_2$;

- a. $k_2 \rightarrow \text{urm} = k_3 \rightarrow \text{urm}$; b. $k_2 \rightarrow \text{urm} = k_3$;
c. $k_2 \rightarrow \text{urm} = k_1 \rightarrow \text{urm}$; d. $k_2 = k_3 \rightarrow \text{urm}$;

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Se consideră arborele cu 6 noduri, numerotate de la 1 la 6, cu muchiile $[2, 1]$, $[2, 4]$, $[4, 5]$, $[6, 2]$, $[6, 3]$. Scrieți toate nodurile desemnate ca rădăcină astfel încât fiecare arbore cu rădăcină obținut să aibă exact 3 frunze. (6p.)
4. Se consideră declarația `char e[20]="51+73"`; Care este șirul memorat de variabila `e` după executarea instrucțiunii de mai jos?
`strcpy(e, strchr(e, '+')+2);` (6p.)
5. Scrieți un program C/C++ care citește de la tastatură un număr natural n ($1 \leq n \leq 100$) și apoi elementele unui tablou bidimensional cu n linii și n coloane, care memorează numere naturale nenule cu cel mult 9 cifre fiecare; programul afișează pe ecran acele valori din tablou care sunt strict mai mici decât toate elementele cu care se învecinează direct (aflate pe aceeași linie dar pe o coloană alăturată sau pe aceeași coloană dar pe o linie alăturată), ca în exemplu. Numerele afișate vor fi separate prin câte un spațiu.
Exemplu: pentru $n=4$ și tabloul alăturat se afișează numerele: 2 0 (2 se învecinează direct cu 4, 3, 6 și 9, și este mai mic decât acestea, iar 0 se învecinează direct cu 6, 9 și 1 și este mai mic decât acestea). (10p.)

5	4	7	9
6	2	3	4
0	9	8	5
1	3	8	6

Subiectul III (30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Aplicând metoda backtracking pentru a genera toate permutările celor n elemente ale unei mulțimi, o soluție se memorează sub forma unui tablou unidimensional x_1, x_2, \dots, x_n . Dacă sunt deja generate valori pentru componentele x_1, x_2, \dots, x_{k-1} , iar pentru componenta curentă, x_k ($1 < k < n$), a fost găsită o valoare convenabilă, atunci se încearcă alegerea (4p.)
- a. unei noi valori pentru componenta x_{k-1} b. unei valori pentru componenta x_{k+1}
 c. unei noi valori pentru componenta x_k d. unei noi valori pentru componenta x_1

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Ce afișează subprogramul `F`, descris alăturat, la apelul `F(5)`? (6p.)
- ```
void F(int x)
{
 cout<<x; | printf("%d", x);
 if(x>=3)
 F(x-2);
 cout<<x-1; | printf("%d", x-1);
}
```
3. Scrieți definiția completă a subprogramului `divizor`, cu trei parametri, prin care primește 3 numere naturale cu cel mult 9 cifre fiecare și returnează numărul divizorilor comuni tuturor celor 3 numere.
4. Se consideră un șir  $s$  format după regula alăturată, unde s-a notat cu  $a \Theta b$  numărul obținut prin concatenarea cifrelor lui  $a$  și  $b$ , în această ordine.  
**Exemplu:** pentru  $x=2$  se obține șirul:  
 2, 3, 32, 323, 32332,.....
- $$s_n = \begin{cases} x & \text{dacă } n=1 \\ x+1 & \text{dacă } n=2 \\ s_{n-1} \Theta s_{n-2} & \text{dacă } n>2 \end{cases}$$
- Fișierul text `SIR.TXT` conține pe prima linie două numere,  $x$  ( $1 \leq x \leq 20$ ) și  $k$  ( $1 \leq k \leq 5000$ ), separate printr-un spațiu, iar pe a doua linie un număr format din exact  $k$  cifre, reprezentând un termen al șirului  $s$  (diferit de  $x$ ). Cifrele numărului nu sunt separate prin spații.
- a) Scrieți un program C/C++ care, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, afișează pe ecran acel termen din șir care îl precede pe cel citit din fișier.

**Exemplu:** dacă fișierul conține valorile alăturate, se va afișa pe ecran numărul 323. (6p.)

|       |   |
|-------|---|
| 2     | 5 |
| 32332 |   |

- b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)

## MODEL DE REZOLVARE

### SUBIECTUL I

- 1.a
2. a. Algoritmul afișează valoarea 1 (cifra de control a nr. a)
- b. Valorile căutate sunt 106 și 997
- c.

```
#include<iostream.h>
#include<conio.h>
unsigned long a;
unsigned b;
void main()
{ clrscr();
 cout<<" a="; cin>>a;
 do{ b=0;
 while(a) { b+=a%10;
 a/=10;
 }
 a=b;
 }
 while(a>=10);
 cout<<b;
 getch();
}
```

d. Numărul a are maxim 10 cifre, prin urmare suma cifrelor are maxim 2 cifre. Pentru noul a sunt suficiente maxim 2 adunări pentru cifrele sale astfel încât să se obțină cifra de control. Deci algoritmul poate fi scris în următoarea formă utilizând o singură structură repetitivă:

```
citește a
b=0
cât timp a>0 executa b ← b+ a%10
 a←a/10

sf. cât timp
a=b
dacă a>=10 atunci b←a%10 + a/10
sf dacă
a=b
dacă a>=10 atunci b←a%10 + a/10
sf dacă
scrie b
```

### SUBIECTUL II

1. d
2. a
3. 2,4,6
4. Șirul de caractere obținut va fi „,3”

5. Parcurgem matricea și pentru fiecare element verificăm proprietatea dată în enunț. Pentru ca să nu avem probleme cu elementele de pe coroana matricii am bordat matricea cu elemente egale cu maximum elementelor matricii. Am utilizat un subprogram test care are ca parametri linia și coloana unui element oarecare din matrice și care returnează valoarea 1 dacă elementul are proprietatea dată și 0 în caz contrar.

```
#include<iostream.h>
#include<conio.h>
long a[100][100],n,i,j,max;
void citire()
{cout<<"n="; cin>>n;
 for(i=1;i<=n;i++)
 for(j=1;j<=n;j++)
 {cin>>a[i][j];
 if(a[i][j]>max) max=a[i][j];
 }
 for(i=0;i<=n+1;i++)
 a[i][0]=a[i][n+1]=a[0][i]=a[0][n+1]=max;
// bordarea
}
```

```
int test(int x,int y)
{ if(a[x][y]<a[x-1][y] && a[x][y]<a[x+1][y])
 if(a[x][y]<a[x][y-1] && a[x][y]<a[x][y+1])
 return 1;
 return 0;
}
void main()
{ clrscr();
 citire();
 for(i=1;i<=n;i++)
 for(j=1;j<=n;j++)
 if(test(i,j)) cout<<a[i][j]<<" ";
 getch();
}
```

**SUBIECTUL III**

1. b

2. Algoritm recursiv dat afișează șirul de valori : 531024.

3. Cel mai mare divizor posibil al celor trei numere poate fi minimul lor (ex. dacă nr sunt 12, 36, 120 cel mai mare divizor comun este 12). De aceea vom calcula minimul celor 3 numere iar divizorul comun poate fi acest minim sau orice numar cuprins între 1 și jumătatea minimului.

```
long divizor(long a, long b, long c)
{ int nr=1;
 long min=a;
 if(min>b) min=b;
 if(min>c) min=c;
 if(a%min==0 && b%min==0 && c%min==0) nr++;
 for(long d=2; d<=min/2; d++)
 if(a%d==0 && b%d==0 && c%d==0) nr++;
 return nr;
}
```

4. Se observă că dacă  $x$  are o cifră atunci numărul de elemente din șirul obținut prin operația de „concatenare” are la fiecare pas un număr de elemente egal cu numărul din șirul lui Fibonacci atașat (1, 1, 1+1, 1+2..) iar dacă  $x$  are două cifre atunci numărul de perechi de caractere coincide cu numărul lui Fibonacci. Deci șirul anterior, pe care noi trebuie să-l afișăm va avea atâtea caractere sau atâtea perechi cât este elementul anterior lui  $k$  din șirul lui Fibonacci sau cât este elementul anterior lui  $k/2$  din același șir. Cu ajutorul unei funcții determinăm acest anterior și afișăm din fișier numărul corespunzător de caractere.

Algoritmul este eficient deoarece nu am utilizat structuri de date iar din fișier nu am afișat decât câte caractere au fost necesare.

```
#include<fstream.h>
#include<conio.h>
int x,k,a,b,c;
char ch;
int precedent(int k)
{ a=0;b=1;c=a+b;
 while(c<k)
 { a=b; b=c; c=a+b; }
 return b;
}
void main()
{ ifstream f("sir.txt");
 f>>x>>k;
 if(x>=9) {int p=precedent(k);
 for(int i=1;i<=p;i++)
 { f>>ch;
 cout<<ch;}
 }
 else {int p=precedent(k/2);
 for(int i=1;i<=p*2;i++) { f>>ch; cout<<ch;}
 }
 f.close();
}
```

## **PERMISUL EUROPEAN DE CONDUCERE A COMPUTERULUI**

*prof. Florentina Ungureanu*  
*prof. Adina Șalaru*

Programul ECDL (European Computer Driving Licence) se bucură de recunoaștere internațională în 146 de țări și este cel mai răspândit program de certificare a cunoștințelor de utilizare a computerului (adoptat și recunoscut de Consiliul European, prin Comisia ESDIS-Employment care recomandă la angajare ca programul ECDL "să fie acceptat drept standard de certificare IT pan-european la nivel de bază."). ECDL este pentru domeniul IT similar cu certificări de genul Toefl sau Cambridge pentru atestarea cunoștințelor de limbi străine.

În România, obținerea Permisului ECDL este recomandată de Asociația Oamenilor de Afaceri și cerută de principalele organizații sindicale, inclusă în programa universitară a numeroase universități, recomandată pentru angajații din administrația publică prin HG 1007/2001 și utilizat în principalele programe de reconversie profesională.

Permisul ECDL atestă faptul ca deținătorul a promovat 7 module: un test teoretic care evaluează *Conceptele de bază ale tehnologiei informației* și șase teste practice care evaluează competența de bază a deținătorului în utilizarea unui computer personal și în folosirea aplicațiilor întâlnite în activitățile curente (*Utilizarea computerului și organizarea fișierelor; Procesare de text; Calcul tabelar; Baze de date; Prezentări; Informație și comunicare*). Pentru cele 7 examene, candidatul trebuie să achiziționeze un Card de Aptitudini ECDL de la **centrul de atestare acreditat** al liceului nostru, **Colegiul Național de Informatică**, sau de la **ECDL România**. Pe acest Card de aptitudini ECDL sunt trecute rezultatele celor șapte examene. Când toate cele șapte module sunt completate, candidatul primește Permisul european de conducere a computerului (ECDL). Cele șapte examene ECDL pot fi luate într-o perioadă de maximum 3 ani, începând cu data susținerii primului examen, ele putând fi date de mai multe ori în această perioadă, dacă este cazul, fără ca eventualul insucces să fie sancționat în alt mod decât prin plata pentru susținerea unui nou examen. Se pot da mai multe teste în aceeași zi în ce ordine dorește candidatul.

La cererea candidaților, după promovarea a patru examene (corespunzătoare modulelor 2 – Utilizarea Computerului și organizarea Fișierelor, 3 – Procesare de text, 7 – Informație și Comunicare și a unui modul la alegere dintre celelalte 4 module) se eliberează și un "Permis european de conducere a computerului Start", care atestă absolvirea a patru module. ECDL Start este acceptat de majoritatea angajatorilor de profil european, acolo unde sunt suficiente cunoștințele dobândite prin parcurgerea a numai patru module.

Informații despre ECDL găsiți și pe site-ul Colegiului Național de Informatică.

Pentru a înțelege mai bine în ce constă examinarea ECDL exemplificăm simulări pentru modulele 1, 3 și 7.

### **Modulul 1 – Conceptele de bază ale tehnologiei informației**

- 1) Tehnologia informației poate fi caracterizată prin:
  - a) normele și procedeele de colectare, memorare, transmitere și prelucrare a datelor, în vederea obținerii rezultatelor cu ajutorul calculatorului electronic
  - b) normele și tehnicile de securitate a muncii
  - c) partea tehnologică a informației
  - d) cuprinde tehnici și procedee de programare
- 2) Minicalculatoarele
  - a) sunt mai mici decât microcalculatoarele
  - b) sunt echivalente microcalculatoarelor din p.d.v. al capacității de stocare a datelor și puterii de calcul
  - c) sunt inferioare microcalculatoarelor din p.d.v. menționate la b)
  - d) sunt superioare microcalculatoarelor din p.d.v. menționate la b)
- 3) Calculatoarele „mainframe”
  - a) din punct de vedere al performanțelor sunt situate între microcalculatoare și supercalculatoare
  - b) nu există acest tip de calculatoare
  - c) sunt calculatoare a căror ecran este împărțit în mai multe ferestre
  - d) sunt calculatoare principale
- 4) Denumirea de microcalculator provine de la:
  - a) micron b) microprocesor c) micro d) microfon
- 5) UCC reprezintă
  - a) Unitatea Centrală de Calcul b) Unitatea de Comandă și Control
  - c) Unitatea Centrală de Comandă d) Unitatea Centrală de Control
- 6) UCP prescurtează:
  - a) Unitatea Centrală de Prelucrare b) Unitatea de Calcul a Proceselor
  - c) Unitatea de Calcul și Prelucrare d) Unitatea de Calculator Personal
- 7) Cea mai mică unitate de memorie este:
  - a) bitul b) byte-ul c) cuvântul d) octetul
- 8) Un bit
  - a) este format din 8 byte b) poate păstra o cifră 0-9
  - c) este un circuit d) poate lua valorile 0 și 1
- 9) Ordonăți descrescător multiplii byte-ului:
  - a) PB,EB,TB,GB,MB,KB b) EB,PB,TB,GB,MB,KB c) EB,PB,GB,TB,KB,MB d) PB,EB,GB,MB,TB,KB
- 10) Interfețele sunt:
  - a) componente ce asigură conectarea unei componente fizice la placa de bază

- b) programe de legătură între echipamente c) programe de instalare a echipamentelor fizice  
d) componente ce asigură o conexiune între 2 elemente pentru ca acestea să poată lucra împreună
- 11) Un byte= a) 1 bit b) 8 biți c) 8 octeți d) 10 octeți
- 12) Care dintre următoarele dispozitive nu sunt dispozitive periferice de intrare:  
a) mouse b) touchpad c) light pen d) plotter
- 13) Care dintre următoarele dispozitive nu sunt dispozitive de ieșire:  
a) Imprimanta b) Display Screen c) VDU d) touchpad
- 14) Dispozitive de intrare/ieșire:  
a) VDU, Touch screen b) Modem, Touchpad c) VDU, Modem d) Modem, Touch screen
- 15) Diagonala monitorului se măsoară în a) cm b) m c) inch d) pixeli
- 16) Touchpad  
a) reprezintă covorașul pentru mouse b) ecranul monitorului  
c) zona de ecran pe care putem puncta cu ajutorul degetului  
d) o suprafață sensibilă la atingere, folosită ca dispozitiv de indicare a poziției
- 17) Ce evenimente nu au loc în cazul unei operații de formatare:  
a) ștergerea datelor existente b) scrierea informațiilor necesare pentru adresarea suportului  
c) marcarea zonelor defecte d) redistribuirea zonelor defecte
- 18) Software-ul de bază reprezintă a) software-ul cu care se lucrează în acel moment  
b) programele de sistem c) programele de aplicații d) software-ul instalat inițial
- 19) Sistemul de operare reprezintă  
a) interfața între componentele hard și utilizator b) interfața serială și interfața paralelă  
c) programele rulate pe sistemul de calcul d) interfața între componentele hard
- 20) Care asociere nu este corectă?  
a) LAN – Local Area Network b) MAN – Metropolitan Area Network  
c) WAN – Word Area Network d) GAN – Global Area Network
- 21) Internetul este o rețea: a) LAN b) GAN c) MAN d) WAN
- 22) În cazul rețelelor, topologia reprezintă  
a) modul în care sunt conectate calculatoarele b) distanța dintre mai multe rețele  
c) aranjarea geometrică a sistemului de calculatoare d) poziția serverului de rețea
- 23) Intranet-ul reprezintă  
a) o parte a Internetului b) o rețea de comunicare privată, internă unei companii  
c) o extensie a Internetului d) o rețea de calculatoare ce nu comunică cu exteriorul
- 24) Extranet-ul este  
a) o parte a Internetului b) o rețea de comunicare privată, internă unei companii  
c) o extensie a unei rețele Intranet d) o rețea de calculatoare ce permite accesul în exterior
- 25) www = a) World Wide Web c) Wide World Web b) Word Wide Web d) Wide Word Web
- 26) Care din următoarele caracteristici nu reprezintă un avantaj oferit de poșta electronică?  
a) flexibilitate b) gestiunea mesajelor c) spațiul de stocare fix d) viteza de transmisie mare
- 27) Dezavantaje ale utilizării comerțului electronic:  
a) nu este necesară deplasarea prin magazine b) acces non-stop  
c) nu pot fi identificați parametrii calitativi ai produselor d) produse primite direct acasă
- 28) Virușii de calculator sunt: a) pictograme pe desktop b) programe create cu scop distructiv  
c) programe create pentru testarea hard-ului d) programe create pentru testarea soft-ului
- 29) Care dintre următoarele exemple reprezintă modalitatea de pregătire cu ajutorul calculatorului?  
a) CAD b) CBT c) CAM d) LAN
- 30) Care dintre următoarele afirmații reprezintă avantajul principal al unui sistem de rețea?  
a) creșterea vitezei de prelucrare a datelor b) costuri reduse  
c) partajarea datelor d) crearea de copii de siguranță a datelor
- 31) Pentru a găsi diferite informații pe Internet utilizați:  
a) un dicționar online b) un motor de căutare c) un program de poștă electronică d) o bază de date
- 32) Majoritatea sistemelor informatice folosesc pictograme și un mouse. Cum se numește acest sistem?  
a) Windows b) Symbolic User Interface c) Graphical User Interface d) Windows NT
- 33) Utilizarea computerului poate duce la leziuni cauzate de mișcări repetate. Care dintre următoarele afirmații este cea mai bună modalitate de a evita acest lucru?  
a) plasarea monitorului mai departe de ochi b) folosirea unei tastaturi ergonomice  
c) lumina să nu bată în monitor d) folosirea unui scaun fără brațe de sprijin.
- 34) Un dispozitiv de intrare este:  
a) dispozitivul de alimentare la curent electric b) dispozitivul de intrare în rețea cu user și parolă  
c) dispozitiv cu ajutorul căruia introduceți date în calculator  
d) dispozitivul cu ajutorul căruia se obțin rezultatele prelucrării
- 35) Dreptul de utilizare al unei aplicații software se numește:  
a) licență b) patent c) copyright d) autorizare
- 36) Modalitatea legală de protejare a lucrărilor literale, științifice, artistice sau de orice fel se numește:  
a) licență b) patent c) copyright d) autorizare

### Modulul 3 – Procesare de text

- 1) Deschideți o aplicație de procesare de text și un document nou pe care să-l denumiți cu numele dumneavoastră.
- 2) Scrieți ca titlu în acest document următorul text: **Simulare Word** și centrați titlul.
- 3) Formatați documentul astfel: dimensiunea hârtiei A4, margini de 1,5 cm sus, jos, dreapta și 3 cm în stânga, font Calibri 11.

**Pe prima pagină** scrieți o adeverință prin care să fie atestat faptul că dumneavoastră sunteți elev/profesor. Adeverința trebuie să respecte următoarele condiții:

- 4) este scrisă pe jumătate de pagină
- 5) în partea stângă – sus apare antetul școlii care eliberează adeverința (numele școlii, adresa, telefon, fax)
- 6) titlul ADEVERINȚĂ este îngroșat și plasat centrat și are adăugată o umbră
- 7) Indentarea paragrafului este de 1 cm pentru primul rând (către dreapta)
- 8) Liniile sunt spațiate la 1,5 rânduri
- 9) Adeverința este semnată de către directorul școlii pe ultimul rând în partea dreaptă.
- 10) Data eliberării este scrisă pe primul rând în partea dreaptă, împreună cu numărul de înregistrare
- 11) Textul este aliniat stânga-dreapta

**În jumătatea a doua a primei pagini** realizați un ecuson pentru o persoană care este elev în școala dumneavoastră după modelul:



#### Pe pagina a doua

- 12) Descrieți cinci caracteristici ale unui bun elev. Fiecare caracteristică are în față un marcator, altul decât cele uzuale
- 13) Inserați un tabel care să cuprindă informații referitoare la cinci dintre colegii dumneavoastră de liceu, după modelul:

| Nr. Crt | Nume Și Prenume Elev | Adresa | Clasa | Observații |
|---------|----------------------|--------|-------|------------|
| 1)      | Albu Crina           |        |       |            |
| 2)      | Cristea Vlad         |        |       |            |
| 3)      | Andrei Emilia        |        |       |            |
| 4)      | Andrei Anca          |        |       |            |
| 5)      |                      |        |       |            |
| 6)      |                      |        |       |            |

- 14) Introduceți datele în tabel și aliniați textul la stânga
- 15) Aliniați textul rândului de antet (al primului rând) la centru atât pe verticală cât și pe orizontală
- 16) Numerotați automat coloana **Nr. Crt** (coloana 1)
- 17) Setări fundalul de umplere a primei linii "Aqua" Fill
- 18) Adăugați bordură de 3 pt tabelului, culoarea Aqua
- 19) Setări liniile de separare a celulelor tabelului ca în modelul de mai sus
- 20) Copiați tabelul și adăugați-l la începutul paginii a doua. Următoarele cinci cerințe se vor realiza pe această copie
- 21) Adăugați și completați o nouă coloană, **medie**, după cea de-a patra coloană (Clasa)
- 22) Ordonăți în ordine descrescătoare după medii, la medii egale în ordine alfabetică
- 23) Redimensionați tabelul astfel încât informațiile să nu fie scrise pe două rânduri
- 24) Setări tabelul astfel încât să nu se permită scrierea unui rând pe pagini diferite
- 25) Transformați literele din prima linie în majuscule
- 26) Imprimați documentul în fișier sub numele **elev.doc**
- 27) Salvați o copie a documentului sub numele **elev2**.
- 28) În subsolul noului document adăugați numărul paginii la stânga, data și ora sistemului la dreapta.
- 29) Salvați documentele în folderul **simulare**, aflat pe Desktop (Dacă nu există, construiți acest folder).

### **Modulul 7 – Informație și comunicare**

1. Unde se tastează adresa paginii web pe care vreți să o încărcați?
2. Care sunt cele două butoane care permit navigarea prin paginile deschise (pagina deschisă anterior și ultima pagină vizitată)?
3. Ce înseamnă accesarea unui URL?
4. Ce este un motor de căutare?
5. Scrieți răspunsurile la primele patru întrebări într-un document de tip Word și salvați-l cu numele **date** în folderul cu numele dumneavoastră.
6. Deschideți o aplicație de tip **motor de căutare**. Căutați informații despre **ECDL**. Scrieți în documentul **date** adresa unei pagini care conține informațiile căutate.
7. Căutați o imagine despre **ECDL** copiați-o în același document. Salvați documentul. Pentru următoarele întrebări scrieți numărul întrebării și un răspuns scurt în același document. Salvați documentul după fiecare modificare.
8. Care sunt cele patru elemente care alcătuiesc structura unei adrese de e-mail?
9. Care dintre următoarele variante reprezintă un avantaj al utilizării e-mail-ului?
  - a) cost scăzut      b) viteză de transmisie mică      c) primirea de mesaje inutile      d) virusare
10. Care dintre următoarele foldere conține mesaje primite de utilizator?
  - a) Sent      b) Inbox      c) Drafts      d) Bulk
11. Cum se deschide un mesaj pentru a putea fi citit?
12. Cum se deschide un fișier atașat unui mesaj primit?
13. Cum se numește caseta de text în care este scris numele și adresa expeditorului?
14. Care este comanda prin care se poate compune un nou mesaj?
15. Care este comanda prin care se răspunde unui mesaj primit?
16. Care este caseta care trebuie completată dacă vrem să trimitem mesajul la mai mulți destinatari?
17. Ce se întâmplă cu adresele scrise în caseta Bcc?
18. Care este caseta de text în care se introduce titlul mesajului?
19. Care sunt etapele de realizare a ștergerii unui mesaj?
20. Cum se poate tipări un mesaj primit?
21. Salvați documentul.
22. Deschideți o aplicație de poștă electronică.
23. Scrieți ca text: data, numele și județul.
24. Atașați mesajului documentul **date** care conține răspunsurile voastre.
25. Atașați mesajului imaginea cu ECDL pe care ați căutat-o pe Internet.
26. Trimiteți mesajul la adresa specificată de examinatori.
27. Deschideți un fișier text nou. Introduceți datele dumneavoastră de identificare pe primul rând al documentului și salvați fișierul sub numele **internet1.doc** pe Desktop.
28. Descrieți pe scurt rolul butonului **Back**, introduceți răspunsul în fișier și salvați.
29. Deschideți o aplicație de navigare pe Internet și accesați adresa: <http://ecdl.org.ro>
30. Copiați sigla ECDL în fișier și salvați.
31. Salvați fișierul în format html (cu numele **internet1.html**).
32. Deschideți o aplicație electronică de e-mail.
33. Scrieți în câmpul **to** adresa specificată de examinatori, iar în câmpul **subject** cuvântul: **Test modul 7**.
34. Atașați fișierele **internet1.doc** și **internet1.html**.
35. Scrieți ca mesaj numele dumneavoastră și trimiteți mesajul.

### **Resurse electronice (simulari ECDL)**

1. <http://www.ecdl.org.ro/download.php>
2. <http://cia.buzau.ro/subiecte.exe>
3. <http://www.telacad.ro/ecdl-start/descriere-curs-ecdl-start.aspx>
4. <http://traianct.liceu.edu.ro/ecdl/teste.php3.htm>
5. <http://www.uvvg.ro/ecdl/simulari.pdf>
6. <http://cni.nt.edu.ro/liceu/ro/ecdl.htm>
7. <http://www.liceulovidius.ro/ecdl/teste%20demo.htm>

## PLATFORMA .NET

prof. Georgeta Nour,  
prof. Elena Costrachevici

.NET este un cadru (*Framework*) de dezvoltare software unitară care permite realizarea, distribuirea și rularea aplicațiilor-desktop Windows și aplicațiilor WEB.

Tehnologia .NET pune laolaltă mai multe tehnologii (ASP, XML, OOP, SOAP, WDSL, UDDI) și limbaje de programare (VB, C++, C#, J#) asigurând totodată atât portabilitatea codului compilat între diferite calculatoare cu sistem Windows, cât și reutilizarea codului în programe, indiferent de limbajul de programare utilizat.

.NET Framework este o componentă livrată împreună cu sistemul de operare Windows. De fapt, .NET 2.0 vine cu Windows Server 2003, se poate instala pe versiunile anterioare, până la Windows 98 inclusiv; .NET 3.0 vine instalat pe Windows Vista și poate fi instalat pe versiunile Windows XP cu SP2 și Windows Server 2003 cu minimum SP1.

Pentru a dezvolta aplicații pe platforma .NET este bine să avem 3 componente esențiale:

- un set de limbaje (C#, Visual Basic .NET, J#, Managed C++, Smalltalk, Perl, Fortran, Cobol, Pascal etc),
- un set de medii de dezvoltare (Visual Studio .NET, Visio),
- o bibliotecă de clase pentru crearea serviciilor Web, aplicațiilor Web și aplicațiilor desktop Windows.

Când dezvoltăm aplicații .NET, putem utiliza:

➤ Servere specializate – un set de servere Enterprise .NET (din familia SQL Server 2000, Exchange 2000 etc.), care pun la dispoziție funcții de stocare a bazelor de date, email, aplicații B2B (*Business to Business* – comerț electronic între partenerii unei afaceri).

➤ Servicii Web (în special comerciale) utile în aplicații care necesită identificarea utilizatorilor (de exemplu, .NET Passport – un mod de autentificare folosind un singur nume și o parolă pentru toate ste-urile vizitate)

➤ Servicii incluse pentru dispozitive non-PC (Pocket PC Phone Edition, Smartphone, Tablet PC, Smart Display, Xbox, set-top boxes, etc.)

### .NET Framework

Componenta .NET Framework stă la baza tehnologiei .NET, este ultima interfață între aplicațiile .NET și sistemul de operare și actualmente conține:

- Limbajele C#, VB.NET, C++ și J#. Pentru a fi integrate în platforma .NET toate aceste limbaje respectă niște specificații OOP numite *Common Type System* (CTS). Ele au ca elemente de bază: clase, interfețe, delegări, tipuri valoare și referință, iar ca mecanisme: moștenire, polimorfism și tratarea excepțiilor.
- Platforma comună de executare a programelor numită *Common Language Runtime* (CLR), utilizată de toate cele 4 limbaje. CTS face parte din CLR.
- Ansamblul de biblioteci necesare în realizarea aplicațiilor desktop sau Web numit *Framework Class Library* (FCL).

### Arhitectura .NET

Componenta .NET Framework este formată din compilatoare, biblioteci și alte executabile utile în rularea aplicațiilor .NET. Fișierele corespunzătoare se află, în general, în directorul WINDOWS\Microsoft\NET\Framework\V2.0... (corespunzător versiunii instalate)

## PROGRAMARE VIZUALĂ

### Concepte de bază ale programării vizuale

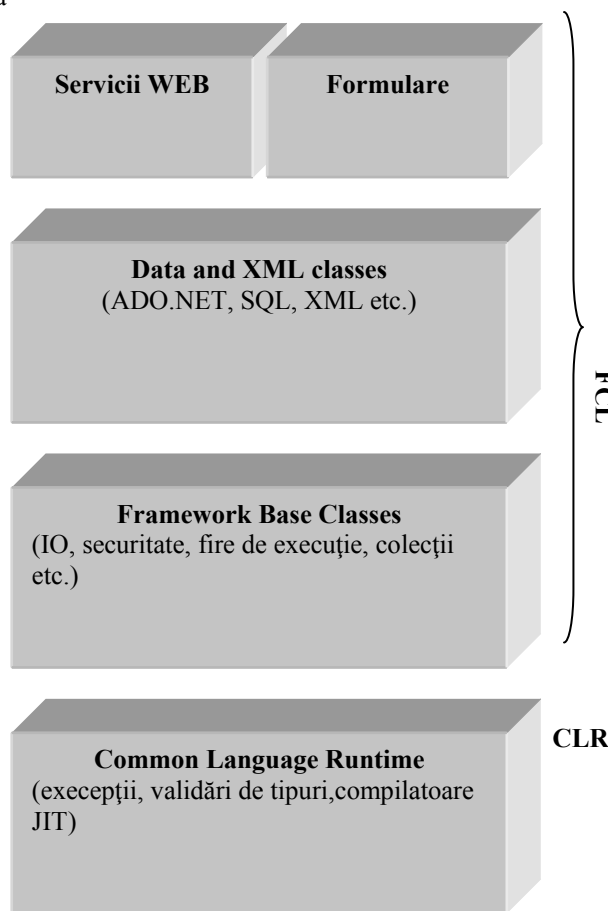
*Programarea vizuală* trebuie privită ca un mod de proiectare a unui program prin operare directă asupra unui set de elemente grafice (de aici vine denumirea de programare vizuală). Această operare are ca efect scrierea automată a unor secvențe de program, secvențe care, împreună cu secvențele scrise textual, vor forma programul.

Spunem că o *aplicație* este *vizuală* dacă dispune de o interfață grafică sugestivă și pune la dispoziția utilizatorului instrumente specifice de utilizare (*drag, click, hint* etc.)

Realizarea unei aplicații vizuale nu constă doar în desenare și aranjare de controale, ci presupune în principal stabilirea unor decizii arhitecturale, decizii ce au la bază unul dintre **modelele arhitecturale** de bază:

#### a) Modelul arhitectural **orientat pe date**.

Acest model nu este orientat pe obiecte, timpul de dezvoltare al unei astfel de aplicații este foarte mic, o parte a codului este generată automat de Visual Studio.Net, codul nu este foarte ușor de întreținut și este recomandat pentru aplicații relativ mici sau cu multe operații de acces (in/out) la o bază de date.



## b) Modelul arhitectural

**Model-view-controller**

Este caracterizat de cele trei concepte de bază: **Model** (reprezentarea datelor se realizează într-o manieră specifică aplicației: conține obiectele de „business”, încapsulează accesul la date), **View** (sunt utilizate elemente de interfață, este format din Form-uri), **Controller** (procesează și răspunde la evenimente iar SO, clasele Form și Control din .Net rutează evenimentul către un „handler”, eveniment tratat în codul din spatele Form-urilor).

c) Modelul arhitectural **Multi-nivel****Nivelul de prezentare ( interfața)**

Se ocupă numai de afișarea informațiilor către utilizator și captarea celor introduse de acesta. Nu cuprinde detalii despre logica aplicației, și cu atât mai mult despre baza de date sau fișierele pe care aceasta le utilizează. Cu alte cuvinte, în cadrul interfeței cu utilizatorul, nu se vor folosi obiecte de tipuri definite de programator, ci numai baza din .NET.

**Nivelul de logică a aplicației**

Se ocupă de tot ceea ce este specific aplicației care se dezvoltă. Aici se efectuează calculele și procesările și se lucrează cu obiecte de tipuri definite de programator.

**Nivelul de acces la date**

Aici rezidă codul care se ocupă cu accesul la baza de date, la fișiere, la alte servicii. Această ultimă structură este foarte bună pentru a organiza aplicațiile, dar nu este ușor de realizat. De exemplu, dacă în interfața cu utilizatorul prezentăm date sub formă *ListView* și la un moment dat clientul ne cere reprezentarea datelor într-un *GridView*, modificările la nivel de cod nu se pot localiza doar în interfață deoarece cele două controale au nevoie de modele de acces la date total diferite.

Indiferent de modelul arhitectural ales, în realizarea aplicației mai trebuie respectate și *principiile proiectării interfețelor*:

• **Simplitatea**

Interfața trebuie să fie cât mai ușor de înțeles și de învățat de către utilizator și să permită acestuia să efectueze operațiile dorite în timp cât mai scurt. În acest sens, este vitală culegerea de informații despre utilizatorii finali ai aplicației și a modului în care aceștia sunt obișnuiți să lucreze.

• **Poziția controalelor**

Locația controalelor dintr-o fereastră trebuie să reflecte importanța relativă și frecvența de utilizare.

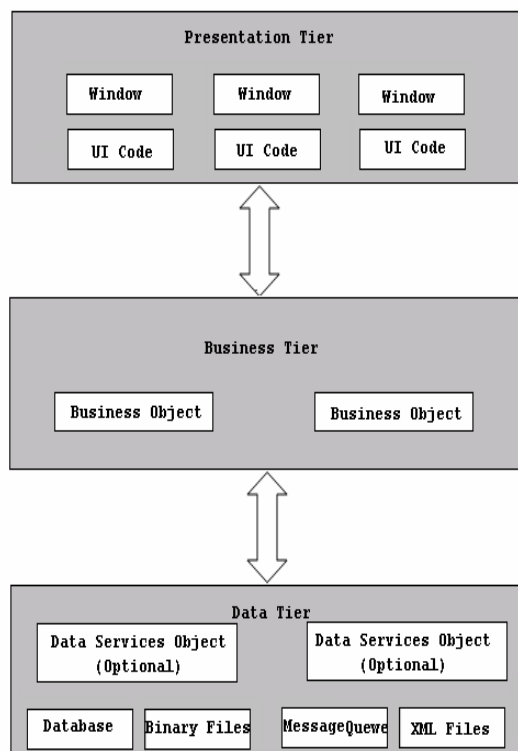
Astfel, când un utilizator trebuie să introducă niște informații – unele obligatorii și altele opționale – este indicat să organizăm controalele astfel încât primele să fie cele care preiau informații obligatorii.

• **Consistența**

Ferestrele și controalele trebuie să fie afișate după un design asemănător („template”) pe parcursul utilizării aplicației. Înainte de a implementa interfața, trebuie decidem cum va arăta aceasta, să definim „template”-ul.

• **Estetica**

Interfața trebuie să fie pe cât posibil plăcută și atrăgătoare.

**Aplicații:**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Timers;
```

```
namespace WindowsApplication1
{
 public partial class Calculare_timpi : Form
 {
 int[] vector = new int[10000];
 int[] copie = new int[10000];
 int[] poz = new int[10000];
 int i,aux;
 DateTime timp_start, timp_sfarsit;
 TimeSpan durata;
 Random r = new Random();
 public Calculare_timpi()
 }
}
```

```

{
 InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
 for (i = 0; i < 10000; i++)
 {
 vector[i] = r.Next(0,50000);
 }
 MessageBox.Show(this.groupBox1,"A fost generat un vector cu " + i + " elemente aleatoare intregi intre 0 si 50000");
 button1.Enabled = false;
 button2.Enabled = true;
 button3.Enabled = true;
 button4.Enabled = true;
 button5.Enabled = true;
 button6.Enabled = true;
}

 private void button2_Click(object sender, EventArgs e)
{
 for (int i = 0; i < 10000; i++)
 {
 copie[i] = vector[i];
 }
 timp_start = DateTime.Now;
 for (int i = 0; i < 9999; i++)
 {
 for (int j = i+1; j < 10000; j++)
 {
 if (copie[i]>copie[j])
 {
 aux = copie[i];
 copie[i]=copie[j];
 copie[j] = aux;
 }
 }
 }
 timp_sfarsit = DateTime.Now;
 durata = timp_sfarsit - timp_start;
 MessageBox.Show("Vectorul a fost sortat prin metoda " + button2.Text + " in " + durata.Milliseconds + "
milisecunde!");
 label1.Text = "aproximativ " + durata.Milliseconds.ToString() + " milisecunde!";
 button2.Enabled = false;
}

private void button3_Click(object sender, EventArgs e)
{
 for (int i = 0; i < 10000; i++)
 {
 copie[i] = vector[i];
 poz[i] = 0;
 }

 timp_start = DateTime.Now;
 for (int i = 0; i < 9999; i++)
 {
 for (int j = i + 1; j < 10000; j++)
 {
 if (copie[j] < copie[i]) poz[i]++;
 else poz[j]++;
 }
 }
 for (i = 0; i < 10000; i++)
 for (int j = 0; j < 10000; j++)
 if (poz[j] == i) copie[i]=vector[j];
 timp_sfarsit = DateTime.Now;
 durata = timp_sfarsit - timp_start;
 MessageBox.Show("Vectorul a fost sortat prin metoda " + button3.Text + " in " + durata.Milliseconds + "
milisecunde!");
 label2.Text = "aproximativ " + durata.Milliseconds.ToString() + " milisecunde!";
 button3.Enabled = false;
}

private void button4_Click(object sender, EventArgs e)
{
 int j,elem,poz;
 for (int i = 0; i < 10000; i++)
 {
 copie[i] = vector[i];
 }
}

```

```

 }
 timp_start = DateTime.Now;
 for (i = 1; i < 10000; i++)
 {
 elem = copie[i];
 j = i - 1;
 while (j >= 0 && copie[j] > copie[i])
 j--;
 poz = j + 1;
 for (j = i; j > poz + 1; j--)
 copie[j] = copie[j - 1];
 copie[poz] = elem;
 }
 timp_sfarsit = DateTime.Now;
 durata = timp_sfarsit - timp_start;
 MessageBox.Show("Vectorul a fost sortat prin metoda " + button4.Text + " in " + durata.Milliseconds + "
milisecunde!");
 label3.Text = "aproximativ " + durata.Milliseconds.ToString() + " milisecunde!";
 button4.Enabled = false;
}

private void button5_Click(object sender, EventArgs e)
{
 for (int i = 0; i < 10000; i++)
 copie[i] = vector[i];
 timp_start = DateTime.Now;
 divimp(0, 9998);
 timp_sfarsit = DateTime.Now;
 durata = timp_sfarsit - timp_start;
 MessageBox.Show("Vectorul a fost sortat prin metoda " + button5.Text + " in " + durata.Milliseconds + "
milisecunde!");
 label4.Text = "aproximativ " + durata.Milliseconds.ToString() + " milisecunde!";
 button5.Enabled = false;
}

private void interclas(int i, int m, int j)
{
 int[] b=new int[10000];
 int x = i;
 int k = 1;
 int y = m + 1;
 while (x <= m && y <= j)
 if (copie[x] < copie[y])
 b[k++] = copie[x++];
 else
 b[k++] = copie[y++];
 while (x <= m)
 b[k++] = copie[x++];
 while (y <= j)
 b[k++] = copie[y++];
 int t = i;
 for (k = 1; k <= (j - i) + 1; k++)
 copie[t++] = b[k];
}

private void divimp(int i, int j)
{if (i < j)
 {
 int m = (i + j) / 2;
 divimp(i, m);
 divimp(m + 1, j);
 interclas(i, m, j);
 }
}

private void button6_Click(object sender, EventArgs e)
{

```

```

for (int i = 0; i < 10000; i++)
{
 copie[i] = vector[i];
}
timp_start = DateTime.Now;
int poz,x;
for (int i = 0; i < 9999; i++)
{
 x = copie[i]; poz = i;
 for (int j = i + 1; j < 10000; j++)
 if (copie[j] < x)
 {
 x = copie[j];
 poz = j;
 }
 copie[poz] = copie[i];
 copie[i] = x;
}
timp_sfarsit = DateTime.Now;
durata = timp_sfarsit - timp_start;
MessageBox.Show("Vectorul a fost sortat prin metoda " + button6.Text + " in " + durata.Milliseconds + "
milisecunde!");
label5.Text = "aproximativ " + durata.Milliseconds.ToString() + " milisecunde!";
button6.Enabled = false;
}
}
}
}

```

## TEHNICI BAZATE PE REPREZENTAREA DATELOR

*prof. Maria Podbereschi*

### Operatori de biți

Operatorii pe biți oferă acces direct la reprezentarea binară a datelor în memorie. Ei pot fi aplicați doar operanzilor de tip întreg cu sau fără semn.

În afară de operatorul `~` care este unar, toți operatorii C prezentați în tabelul de mai jos sunt binari.

| DENUMIRE OPERATOR    | OPERATOR IC/C++       | FORMULĂ          | MOD DE FUNCȚIONARE                                                                                                                                         |
|----------------------|-----------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ȘI                   | <code>&amp;</code>    | $a \& b = 1$     | Dacă $a=b=1$ . Altfel 0                                                                                                                                    |
| COMPLEMENTARE        | <code>~</code>        | $\sim a = 0$     | Dacă $a=1$ , altfel 0                                                                                                                                      |
| SAU                  | <code> </code>        | $a   b = 0$      | Dacă $a=b=0$ . Altfel 1                                                                                                                                    |
| SAU EXCLUSIV         | <code>^</code>        | $a \wedge b = 1$ | Dacă a și b au valori diferite, altfel 0                                                                                                                   |
| DEPLASARE LA STÂNGA  | <code>&lt;&lt;</code> | $c \ll d$        | Operatorul <code>&lt;&lt;</code> realizează deplasarea la stanga pentru operatorul din stanga (c), cu numărul de poziții dat de operandul din dreapta (d). |
| DEPLASARE LA DREAPTA | <code>&gt;&gt;</code> | $c \gg d$        | Operatorul <code>&gt;&gt;</code> realizează deplasarea la dreapta pentru operandul din stanga (c), cu numărul de poziții dat de operandul din dreapta (d). |

**OBSERVAȚIE:** a și b sunt biți iar c și d sunt numere întregi.

**Exemplul 1.** Afișați produsul a două numere x, y folosind algoritmul a la russe.

Rezolvare. Se inițializează produsul cu 0. Se parcurge reprezentarea binară a lui x de la poziția cea mai nesemnificativă la poziția cea mai semnificativă și dacă cifra binară curentă i este 1, atunci cel de-al doilea număr y se înmulțește cu  $2^i$  și se adună la valoarea produsului.

```

unsigned int produs (unsigned int x, unsigned int y)
{
 unsigned int p;
 for(p=0;x!=0;x>>=1,y<<=1)
 if(x&1)
 p+=y;
 return p; }

```

**Exemplul 2.** Fie  $n$  un număr natural scris în baza 10 și  $b$  ( $2 \leq b \leq 256$ ) o baza de numerație. Să se scrie un program care afișează în baza 10 toate numerele mai mici sau egale cu  $n$ , care scrise în baza  $b$  folosesc numai cifrele 0 și 1. Numerele vor fi scrise în ordine strict crescătoare.

**EX:**  $n=11$ ,  $b=3$                       R: 0, 1, 3, 4, 9, 10

Rezolvare: Se va opta pentru reprezentarea numerelor mai mici ca  $n$  în baza 2 urmată de calcularea numerelor în baza 10. unsigned int numar (unsigned int n, unsigned int b)

```
{
int i,nr,pbl;
for(i=2; ; i++)
{ for(nr=0, x=1, pb=1;x;>=1,bp*=b)
nr+=(x&1)*pb;
if(nr<=n) cout<<nr<<endl;
else break; } }
```

### Deplasări pe biți

Deplasările pe biți (shiftări) calculează de fapt înmulțirile ( $\ll$ ) cu 2 și împărțirile ( $\gg$ ) la 2 pentru un număr dat.

| C/C++     | Efect        |
|-----------|--------------|
| $A \ll B$ | $A * 2^B$    |
| $A \gg B$ | $A * 2^{-B}$ |

Operațiile de deplasare pe biți vor fi folosite în locul împărțirilor pentru a îmbunătăți timpul de acces. Deoarece biții sunt mutați către un capăt, la celălalt capăt se adaugă zerouri. O deplasare nu este o rotație, biții deplasați dincolo de capăt fiind pierduți.

O astfel de operație este cel puțin la fel de rapidă ca o adunare (dacă stăm și ne gândim la modul de funcționare pe biți). Astfel costul obținerii "segmentului" ( $X \gg 3 = X/8$ ) în care se află valoarea lui  $X$  este echivalent cu o adunare.

Într-o implementare, pentru a găsi bitul asociat unui număr dat  $X$  avem nevoie de numărul grupei și de poziția bitului în cadrul grupei. În cazul în care implementăm hărțile de biți folosind vectori de tipul unsigned int (16 biți), numărul grupei este dat de formula  $X \gg 4$  iar în cazul în care folosim tipul unsigned long int (32 biți), numărul grupei este dat de formula  $X \gg 5$ .

OBS:  $1 \ll 15$  setează bitul cel mai semnificativ și calculează cea mai mare valoare posibilă pe tipul întreg (16 biți) și anume 32768.

### Măști de biți

Acum știm cum să stabilim poziția unui element într-un vector de biți, dar trebuie să știm și cum să accesăm un bit dintr-un vector de biți. Pentru acest lucru folosim măștile de biți. O mască este o variabilă care aplicată valorii ce conține informația căutată (starea unui bit sau a mai multor biți) returnează starea acesteia: 0 sau 1. Aplicarea se face folosind operatorul  $\&$  pe biți.

### Problema $A^B \bmod C$

Se dau trei numere natural  $A, B, C$ . Să se calculeze numărul  $A^B \bmod C$ .

#### Date de intrare:

În fișierul de intrare abc.in se vor găsi pe prima linie cele trei numere  $A, B, C$ .

#### Date de ieșire:

În fișierul de ieșire abc.out se va afișa numărul cerut în enunț.

#### Restricții:

$0 < A, B, C < 2\,000\,000\,000$

Timp de execuție pe test: 0.1 secunde

#### Soluție:

Vom considera reprezentarea binară a numărului  $B$ . Folosind formula de recurență:

$$[A^B] = \begin{cases} [A^{B/2}]^2, & \text{dacă } B \text{ este par} \\ [A^{B/2}]^2 * A, & \text{dacă } B \text{ este impar} \end{cases}$$

Vom parcurge numărul  $B$  pornind de la bitul cel mai semnificativ spre cel mai nesemnificativ, la fiecare pas ridicând la pătrat numărul și înmulțindu-l cu  $A$  în cazul unui bit 1 întâlnit. Complexitatea algoritmului este  $\log B$ , foarte mică în comparație cu timpul de execuție acordat.

#### Rezolvare:

//O(logB) sau O(1) în soluția aceasta

//Memoria folosită: O(1)

#include<stdio.h>

FILE \*in,\*out;

unsigned long long ABC,A,B,C;

unsigned long i;

int main()

{ in=fopen("abc.in","r");

fscanf(in,"%lld%lld",&A,&B,&C);

fclose(in);

for(ABC=1,i=1;<<31;i;>=1)

if(i&B) ABC=((ABC\*ABC)%C)\*A)%C;

else ABC=(ABC\*ABC)%C;

out=fopen("abc.out","w");

fprint(out,"%u\n",(unsigned long)ABC\_;

fclose(out);

return 0;

}

## ***INTERFEȚE GRAFICE ÎN JAVA***

*prof. Gabriela Blaga*

Java este un mediu de programare ce oferă utilizatorului cadrul necesar și uneltele necesare pentru dezvoltarea aplicațiilor Java. Programele Java pot rula pe diferite tipuri de platforme, cu condiția ca pe aceste platforme să fie instalată o mașină virtuală Java deasupra platformei respective. Avantajul limbajului Java, comparativ cu alte limbaje de programare este conexiunea strânsă cu tehnologia Internet.

### **Caracteristici ale limbajului Java**

- Prezintă concepte fundamentale care sunt deosebit de *simple*. Tehnologia Java pune la dispoziție așa numitul *Garbage Colector*, care face ca programatorul să nu mai aibă grija dealocării zonelor de memorie anterior alocate.
- Spre deosebire de C/C++, este în întregime *orientat pe obiecte*.
- Datorită existenței a două nivele de verificare pentru mărirea gradului de siguranță: unul la compilare (prezent în marea majoritate a limbajelor) și unul în timpul rulării, un program este mai puțin supus erorilor.
- Programele Java pot rula pe orice platformă pentru care există un emulator Java. Partea negativă este că folosirea emulatorului duce la mărirea timpului de execuție.
- Multe din firmele care dezvoltă browsere www au implementat mașina virtuală Java în interiorul acestor browsere. Acesta este unul din avantajele care a făcut limbajul Java așa de popular. Amintim cele mai utilizate browsere care suportă Java: *Netscape Navigator* al firmei *Netscape* și *Internet Explorer* al firmei *Microsoft*.

La baza dezvoltării unui program Java stă mediul de dezvoltare pus la dispoziție de firma Sun. Acesta este *Java Developers Kit (JDK)* și trebuie considerat ca mediu de referință în programarea Java.

Mediul JDK conține pe de o parte o serie de biblioteci de clase Java necesare scrierii unui program și pe de o parte un set de utilitare necesare compilării, testării, execuției și documentării unei aplicații Java.

O clasă este un fișier bytecode cu extensia `.class` și reprezintă unitatea fundamentală a unui program executabil Java. O bibliotecă de clase cuprinde o serie de clase ce au un numitor comun. O astfel de bibliotecă este cunoscută în java sub numele de *package*.

Un package este o colecție de clase și interfețe Java între care există o relație de asociere. În dezvoltarea interfețelor grafice de comunicare cu utilizatorul și dezvoltarea aplicațiilor ce implică grafică cele mai folosite sunt **package java.awt., java.awt.event, package javax.swing**.

JDK-ul conține câteva package-uri fundamentale, package-uri care conțin clase fără de care nu se pot dezvolta aplicații Java. Package-urile Java incluse în JDK formează principalul API- Application Programming Interface.

Orice alt mediu care poate executa aplicații Java diferit de mediul JDK trebuie să includă acest API. Pe lângă aceste pachete fundamentale, utilizatorul poate folosi și altele dezvoltate chiar de el. Trebuie însă să se asigure că aceste pachete adiționale sunt disponibile și pe platforma pe care aplicația se execută, nu numai unde aceasta a fost creată.

În principiu, crearea unei aplicații grafice presupune următoarele lucruri:

- crearea unei suprafețe de afișare (de exemplu o fereastră) pe care vor fi așezate obiectele grafice care servesc la comunicarea cu utilizatorul (butoane, controale de editare, texte, etc);
- crearea și așezarea obiectelor grafice pe suprafața de afișare în pozițiile corespunzătoare;
- definirea unor acțiuni care trebuie să se execute în momentul când utilizatorul interacționează cu obiectele grafice ale aplicației.

Majoritatea obiectelor grafice sunt subclase ale clasei *Component*.

În general, toate componentele sunt definite de clase proprii ce se găsesc în pachetul `java.awt`, clasa *Component* fiind superclasa abstractă a tuturor acestor clase.

Crearea unui obiect grafic nu realizează automat și afișarea lui pe ecran. Ele vor deveni vizibile în momentul în care suprafața pe care sunt afișate va fi vizibilă.

De cele mai multe ori programul trebuie să facă o anumită prelucrare în momentul în care utilizatorul a efectuat o acțiune și, prin urmare, obiectele grafice trebuie să genereze evenimente în funcție de acțiunea pe care au suferit-o. În program trebuie specificat codul care se execută la apariția unui eveniment.

Pachetul de clase *Swing* reprezintă soluția furnizată de firma Sun pentru crearea unor interfețe utilizator grafice complet portabile pe orice platformă.

În *Swing*, toate numele claselor încep cu litera *J*, și atunci când este posibil, numele este același cu cel al clasei *AWT* pe care o înlocuiește.

În *swing* există trei clase pentru gestionarea ferestrelor care sunt:

- *JFrame*,
- *JWindow*
- *JDialog*.

Clasa *JFrame* permite crearea unei ferestre de aplicație. Fereastra are o bara de titlu, o imagine, butoane de minimizare, maximizare și închidere (butoane "system").

Clasa *JWindow* permite crearea unei ferestre fără bara de titlu, meniu, butoane sistem etc.

Clasa *JDialog* permite crearea de ferestre de dialog. Ferestrele de dialog sunt dependente de ferestrele părinte de tip *Frame*.

Crearea unei ferestre principale se realizează cu secvența:

```
public static void main(String args[])
{
```

```
JFrame nume_fer = new JFrame("fereastră");
nume_fer.setSize(200, 200);
nume_fer.show();
}
```

Pentru a accesa conținutul unei ferestre se va folosi funcția `getContentPane()` de forma:

```
Container c = nume_fer.getContentPane();
```

Închiderea automată a aplicației când se apasă butonul Close, se realizează utilizând metoda `setDefaultCloseOperation` astfel:

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Pentru organizarea elementelor într-o fereastră se vor folosi panouri din swing ca: `JPanel` (simple) și `JScrollPane` (cu bare de derulare).

Exemple de componente din biblioteca `javax.swing`

|                     |                                                                                                                                                                                    |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>JButton</b>      | Un simplu buton de comandă care poate fi apăsat printr-un click de mouse                                                                                                           |
| <b>JCheckBox</b>    | Un buton care poate fi "bifat"                                                                                                                                                     |
| <b>JComboBox</b>    | O listă drop-down clasică care poate conține opțional și un câmp textual editabil. Utilizatorul poate de asemenea selecta o valoare din listă care va fi afișată în câmpul textual |
| <b>JFileChooser</b> | O componentă care asigură un mecanism simplu pentru selectarea unui fișier                                                                                                         |
| <b>JLabel</b>       | O componentă care conține un șir textual sau o imagine (nu reacționează la evenimente de introducere/editare text)                                                                 |
| <b>JList</b>        | O componentă care permite utilizatorului să selecteze unul sau mai multe dintre elementele care le conține                                                                         |
| <b>JRadioButton</b> | Un buton de stare on/off. De obicei sunt organizate în grupuri dintre care unul singur poate fi selectat pe poziția "on"                                                           |
| <b>JScrollPane</b>  | O componentă care gestionează o porțiune vizualizabilă prin derularea conținutului în cadrul unui view                                                                             |
| <b>JSlider</b>      | O componentă care permite utilizatorului să selecteze o valoare prin rularea unui indicator pe o scală (asemenea unui "potențiomtru")                                              |
| <b>JTextArea</b>    | O zonă care poate găzdui mai multe linii de text editabile sau nu                                                                                                                  |
| <b>JTextField</b>   | O zonă pentru introducerea unei singure linii de text                                                                                                                              |

*Aplicație: organizarea elementelor într-o fereastră:*

```
package javaapplication6;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class exemplu1 extends JFrame
{
 JLabel jlabel;
 ButtonGroup grup;
 JRadioButton [] butoaneRadio;
 JTextField informatie1 = new JTextField(20);
 JTextField informatie2 = new JTextField(20);
 JButton b1 = new JButton("ButonA");
 JButton b2 = new JButton("ButonB");
 JCheckBox cb1 = new JCheckBox("Optiune1");
 JCheckBox cb2 = new JCheckBox("Optiune2");
 JCheckBox cb3 = new JCheckBox("Optiune3");
 public exemplu1() {
 Container contentPane = getContentPane();
 jlabel = new JLabel();
 jlabel.setIcon(new ImageIcon("C:\\p0.jpg"));
 contentPane.add(jlabel);
 grup = new ButtonGroup();
 butoaneRadio = new JRadioButton[]
 {
 new JRadioButton("imag1"),
 new JRadioButton("imag2"),
 new JRadioButton("imag3"),
 };
 contentPane.setLayout(new FlowLayout());
 for (int i = 0; i < butoaneRadio.length; i++)
 {
 grup.add(butoaneRadio[i]);
 contentPane.add(butoaneRadio[i]);
 butoaneRadio[i].addActionListener(new ActionListener()
 {
 public void actionPerformed(ActionEvent e)
 {
 refacere_imagine(e);
 }
 });
 }
 contentPane.add(b1);
 contentPane.add(b2);
 cb1.setSelected(false);
 cb2.setSelected(false);
 cb3.setSelected(false);
 contentPane.add(cb1);
 contentPane.add(cb2);
 contentPane.add(cb3);
 butoaneRadio[0].setSelected(true);
 contentPane.add(informatie1);
 informatie1.setText("Nu ati selectat nici un buton");
 contentPane.add(informatie2);
 informatie2.setText("Nu ati selectat nici casuta");
 b1.addActionListener(new ActionListener()
 {
 public void actionPerformed(ActionEvent e)
 {
 informatie1.setText("Ai apasat pe butonul A");
 }
 });
 b2.addActionListener(new ActionListener()
 {
 public void actionPerformed(ActionEvent e)
 {
 informatie1.setText("Ai apasat pe butonul B");
 }
 });
 cb1.addActionListener(new ActionListener()
 {
 public void actionPerformed(ActionEvent e)
 {
 if(cb1.isSelected())
 {
 informatie2.setText("Ai selectat casuta 1");
 }
 }
 });
 }
}
```

```

 cb2.setSelected(false);
 cb3.setSelected(false);
 }
}
);
cb2.addActionListener(new ActionListener(){
 public void actionPerformed(ActionEvent e){
 if(cb2.isSelected())
 {informatie2.setText("Ai selectat casuta 2");
 cb1.setSelected(false);
 cb3.setSelected(false);
 }
 }
});
cb3.addActionListener(new ActionListener(){
 public void actionPerformed(ActionEvent e){
 if(cb3.isSelected())
 {informatie2.setText("Ai selectat casuta 3");
 cb1.setSelected(false);
 }
 }
});
 cb2.setSelected(false);
 } } });
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(200, 500);
}

public void refacere_imagine(ItemEvent e){
 if (butoaneRadio[0].isSelected())
 jLabel.setIcon(new ImageIcon("C:\\p1.jpg"));
 if (butoaneRadio[1].isSelected())
 jLabel.setIcon(new ImageIcon("C:\\p2.jpg"));
 if (butoaneRadio[2].isSelected())
 jLabel.setIcon(new ImageIcon("C:\\p3.jpg"));
}

public static void main(String[] args) throws Exception
{
 UIManager.setLookAndFeel("com.sun.java.swing.plaf.
 windows.WindowsLookAndFeel");
 new exemplu1().show();
}

```



**Motto:**

- *Imaginația este mai importantă decât cunoașterea*

**Albert Einstein**

## **MECATRONICA, TEHNOLOGIA VIITORULUI**

*Andone Cosmin, (clasa a XII-a D)*

Mecatronica este un domeniu transdisciplinar al ingineriei, o combinație sinergetică între mecanică de precizie, sistemele electronice de comandă și control și informatică, ce servește proiectării, realizării, punerii în funcțiune și exploatarea de sisteme automate inteligente.

Conceptul de mecatronică s-a născut în Japonia la începutul deceniului al optulea al secolului trecut.

Termenul în sine a fost brevetat de către concernul Yaskawa Electric Co. și a fost utilizat pentru a descrie fuziunea tehnologică.

### **MECANICA, ELECTRONICA, INFORMATICA**

Tot ceea ce numim astăzi produs de înaltă tehnicitate, este produs mecatronic.

Mecatronica este rezultatul evoluției firești în dezvoltarea tehnologică. Tehnologia electronică a stimulat această evoluție. Dezvoltarea microelectronicii a permis integrarea electromecanicii.

În următoarea etapă, prin integrarea microprocesoarelor în structurile electromecanice, acestea devin inteligente și astfel s-a ajuns la mecatronică.

Tehnologia mecatronică aduce în centrul atenției problema informației care, este componenta dăătoare de ton în raport cu materialul și energia. Aceasta poziție a informației este motivată de către japonezi prin următoarele argumente.

- informația asigură satisfacerea nevoilor spirituale ale omului;
- numai informația crește valoarea nou adăugată a tuturor lucrurilor;
- informația înseamnă cultură.

### **EXEMPLE DE PRODUSE ȘI SISTEME MECATRONICE**

Practic tot ceea ce numim produs de înaltă tehnicitate este produs mecatronic. Automobilul modern, roboții, tehnica de calcul, tehnica de telecomunicații, aparatura biomedicală, sistemele de transport inteligent, aparatură de cercetare, aparatură electrocasnică, aparatură cine-foto și audio-video, mașinile agricole moderne etc., sunt exemple reprezentative de produse mecatronice.

Mecatronica s-a născut în mediul industrial. Strădaniile la nivel academic pentru a asigura pregătirea specialiștilor în acord cu cerințele noii tehnologii au condus la conturarea principiilor mecatronice în educație.

Aceste principii sunt:

- hands on: prezența palpabilă a obiectului de studiu;
- learning by doing: învățarea prin practică
- interaction: interacțiunea sistemelor mecanice, electronice, informatice.

Laboratoarele interdisciplinare de mecatronică constituie baza pentru materializarea principiilor: educație prin practică, educație prin cercetare.

Aflată la intersecția unor domenii ale științei cu performanțe de vârf în implementarea noilor tehnologii, mecatronica abordează concepte și sisteme noi în ingineria micro și nano senzorială și sistemelor de acționare, materiale și compozite pretabile pentru implementări la scara celulară sau atomică, structuri celulare și rețele neuronale, sisteme ce prefigurează conceptele de nanoelectronică capabile să producă viitoarele nano-procesoare, noi concepte ale inteligenței artificiale privind adaptabilitatea, capacitatea de a raționa, capacitatea de instruire, noi sisteme de conducere axându-se în special pe controlul robust, tolerant la defecte, adaptiv, inteligent, sisteme expert și neuro-fuzzy etc.

Sistemele mecatronice asigură:

- multifuncționalitate;
- flexibilitate;
- posibilitatea acționării la distanță;
- evoluție continuă datorată dinamicii cerințelor pieței;
- imitare a naturii-adaptabilitate.

Oferind soluții eficiente pentru promovarea interdisciplinarității, mecatronica a devenit suportul demersurilor pentru stimularea inițiativei și a creativității.

Foarte curând mecatronica a devenit filosofie. Pentru practica inginerescă filosofia mecatronică a marcat saltul de la ingineria tradițională, secvențială, la ingineria simultană sau concurrentă.

În ultimii ani mecatronica este definită simplu: știința mașinilor inteligente. Mai recent demersurile pentru înnoire în educație și cercetare aduc în atenție problema mecatronicii ca: mediu educațional în societatea informațională, respectiv mediu de proiectare și fabricare integrată pe fundalul căruia s-a dezvoltat conceptul de proiectare pentru control.

În literatura de specialitate au devenit consacrate extinderi în alte domenii ca: hidronica, pneutronica, termotronica, autotronica, agromecatronica (agricultură de precizie).

Evoluția în dezvoltarea tehnologică înseamnă: micromecatronică, nanomecatronică și biomecatronică. Tendința gene-

rala este de intelectualizare a mașinilor și sistemelor .

## INTEGRONICA ȘI MECATRONICA

Integronica este știința proceselor de integrare și a sistemelor hiperintegrate, așa cum este organismul uman. Ea ține seama de unitatea indisolubilă a lumii în care trăim și necesitatea unei perspective unice asupra acestei lumi.

Unitatea știință, literatură, artă, tehnologie, se realizează în cadrul definit de matematică, cibernetică și filosofie.

## ZOHO – SUITA OFFICE ONLINE

*Băeșu Vlad (clasa a X-a C)*

**Google** încă mai lucrează pentru a termina suita sa office online Google Docs, dar o companie numita AdventNet are deja una, numita **Zoho** și acesta include tot, de la procesare text, foi de calcul, prezentări, notebook la wiki, baze de date online, management de proiect.

**Zoho** a optat pentru o interfață mai familiară, așa că cele mai multe dintre produsele lor sunt inspirate de la Microsoft Office. Spre deosebire de **Google Docs & Spreadsheets** care are o pagină de start, care listează toate fișierele, **Zoho** utilizează o bară laterală care arată fișierele create folosind aplicația activă.

**Zoho Writer** are o interfață cu taburi, astfel că nu trebuie să pleci de la pagina curentă pentru a trece la un nou document. În timp ce multe opțiuni de lucru sunt ca și la **Google Docs**, **Zoho** împinge limitele și vă permite să vă salvați template-uri și chiar le puteți partaja într-un șablon de bibliotecă publice.

**Zoho Writer** vă permite să publicați un document, postându-l la un blog sau să colaborați cu persoanele de contact. **Writer** mai are abilitatea de a exporta documente în noul format Word-docx (acest lucru adaugându-se la suportul pentru fișierele doc, txt, html, pdf, odf, sxw, rtf).

Integrarea între **Zoho's Apps** nu este foarte vizibilă, deci, dacă doriți să treceți de la **Zoho Writer** la **Zoho Sheet**, va trebui să tastați URL-ul în bara de adrese sau să utilizați marcasele.

**Zoho Creator** este un lucru pe care **Google** nu îl are și de fapt. Este versiunea online a Microsoft Access și ea vă permite să creați și să administrați baza de date a cererilor. Cel mai frumos lucru despre **Zoho App** este că puteți construi forme care pot fi încorporate în site-ul dvs. și va popula baza de date. Aveți posibilitatea de a crea forme de contact sau lucruri mai complexe ca o aplicație CRM. Există o mare listă de template-uri, un mod simplu de a construi script-uri de forme, pe care le-ați notifica prin email.

**Zoho** are API-uri, care permit ca unele servicii să poată lista fișierele găzduite de la **Zoho**, la import/export fișiere și, de asemenea, pentru a edita documente de la distanță.

În timp ce în **Google Docs** lipsesc multe caracteristici deja disponibile în **Zoho** caracteristicile existente sunt mai limitate, în **Google Docs** cererile sunt mult mai stabile și lucrează mai bine dacă partajați fișiere cu alte persoane.

**Zoho** oferă o mulțime de instrumente, inclusiv un plugin pentru Microsoft Office, care vă permite să lucrați la fișierele dumneavoastră online și să le salvați la **Zoho** din Microsoft Word și Excel.

Am încercat **Zoho Writer** în browserele Mozilla Firefox 3.0.5, Microsoft Internet Explorer 7, Safari 3, Google Chrome 1, Konqueror și Opera și nu am avut nici o problemă la editarea unui document în nici unul din aceste browsere.

## UN NOU SISTEM DE OPERARE

– UBUNTU –

*Manolachi Silviu  
Tabacariu Ioan  
(clasa a XI-a C)*

Pe scurt: **ubuntu** este un cuvânt vechi de origine africană având multe valențe, ce privesc comportamentul ideal al oamenilor față de cei din jur („bunătațe”, „ajută și vei fi ajutat” etc.). Un om, măcar aparent de tip „ubuntu”, Mark Richard Shuttleworth, a folosit acest cuvânt pentru a da numele unei distribuții Linux cu principii frumoase: să fie și să rămână pe veci 100% gratuită, să poată fi utilă imediat după instalare, să fie de neclintit în fața virusilor, să primească tot gratuit și fără pauze de-un veac actualizări software de securitate. Ba chiar să ajungă la oricine și-o dorește, fără ca acesta să fie nevoit să scoată vreun ban din buzunar.

### Câteva explicații (necesare) despre diferențele Ubuntu (Linux, în general) – Windows

În primul rând pe Ubuntu te așteaptă o altă interfață grafică, combinație între cea de Mac OS X și cea de Windows. E chiar frumoasă, mai ales datorită culorilor și iconițelor plăcute ochiului, modulul inteligent de aranjare a meniurilor/ submeniurilor, poate faptul că e disponibilă și în limba română. Este și o interfață foarte „curată”, în sensul că toate aplicațiile urmează look-ul general, impus de anumite standarde de design și a interfeței cu utilizatorul

Cu programele e marele bai. La prima încercare a Ubuntu sau a oricărui alt sistem de operare de tip Linux, un utilizator obișnuit cu Windows (98, XP, Vista etc.) nu găsește programele pe care le știe. Asta pentru că pe Ubuntu există unele mai proaste, altele mai deștepte, însă toate pe gratis, dar având firește alte denumiri decât cele cu care ești obișnuit. Unele – nu puține – sunt disponibile și pentru Windows: Pidgin (multimessenger pentru Yahoo, MSN și alte servicii de mesagerie instant),

GIMP – similar cu Adobe Photoshop, Firefox, ThunderBird pentru email, OpenOfficeOrg (suita office similară Microsoft Office). Alte câteva zeci de mii de programe sunt disponibile pe net, pentru aproape absolut tot ce vrei să faci pe computer, iar instalarea e foarte simplă. Pentru aproape orice program de Windows există un corespondent în Ubuntu. Gamerii suferă cel mai mult, deoarece nu merg jocurile alea complicate și frumoase cu care s-au obișnuit pe Windows, totuși cu un programel (un așa-numit emulator) destul de costisitor, ce împacă capra cu varza în acest sens ai putea să rezolvi și această problemă.

#### De ce să aleg Ubuntu?

- se lansează o nouă versiune, cu soft la zi, cam la fiecare șase luni;
- îl poți instala alături de Windows, pe o partiție separată; la pornirea computerului selectezi sistemul de operare pe care vrei să-l folosești pentru sesiunea respectivă;
- are susținere financiară din partea Canonical – firmă condusă de un om pasionat de tehnologie, dornic să demonstreze că se poate câștiga și de pe urma oferirii unor produse pe gratis; susținerea financiară presupune în primul rând faptul că o mână de oameni au grijă ca Ubuntu și programele recomandate pentru acest sistem de operare să aibă o anumită "prestanță" (că te poți bizui pe ele);
- sute de servere, de pe toate continentele, găzduiesc cele mai recente pachete software și update-uri de securitate pentru Ubuntu;
- cel mai popular sistem de operare din lume bazat pe Linux cu o comunitate impresionantă, câteva milioane de fani/utilizatori – caută orice te interesează despre Ubuntu pe Google, pe forumurile Ubuntu (vezi <http://forum.ubuntu.com> și, în limba română, <http://forum.ubuntu.ro>);
- interfața grafică intuitivă și plăcută ochiului;
- nu-ți pui problema virușilor;
- singurul soft 150% gratuit: primești și CD pe gratis dacă-l comanzi de pe net

#### Cum fac rost de sistemul ăsta de operare?

Intri pe situl Ubuntu ([www.ubuntu.com](http://www.ubuntu.com)), dai clic pe butonul download. Apoi alegi Ubuntu 8.10 Desktop Edition pentru laptop sau desktop (Server Edition e pentru servere) și, în funcție de tipul de procesor din computerul tău – pe 32 sau 64 de biți, bifezi "Standard personal computer", respectiv "64bit AMD and Intel computers". Selectezi locația, alegi dintr-o gramadă de servere pentru a-l descărca cât de repede. Având 700MB, e bine să folosești un soft de download, de pildă Free Download Manager, ca să nu te trezești cu erori la finalul descărcării. Ceea ce ai descărcat pe hard disk se numește "image ISO", iar cu un soft specializat trebuie ars pe CD. Dacă trăiești pe Marte și n-ai internet sau pur și simplu vrei un CD original, pe gratis, intră pe <http://shipit.ubuntu.com>, fă-ți un cont de utilizator și comandă CD-ul. Vine în vreo două săptămâni prin poștă.

#### Ultima versiune, Ubuntu 8.10 – cum se prezintă.

Instalarea și detectarea automată a driverelor. După ce am descărcat Ubuntu 8.10, l-am inscripționat pe un CD. Apoi, am purces la instalare; durata acesteia variază de la 15 la 40 de minute, în funcție de experiența ta și caracteristicile computerului.

Pașii de urmat sunt șapte, dar cu o singură excepție, banali: selectarea limbii, a orașului, a layout-ului preferat pentru tastatură, partiționarea hard disk-ului (aceasta fiind partea mai complicată/periculoasă pentru un începător), introducerea numelui și a parolei, opțional – importarea unor date de pe Windows (dacă-l ai instalat) și în final să bifezi că totul e ok și se poate începe instalarea propriu-zisă. Legat de partiționare, Ubuntu are nevoie de o partiție pentru sistemul de operare (peste 2GB) și una pe care o utilizează pe post de memorie RAM virtuală, partiție numită SWAP (512MB - 1GB sunt de ajuns). La pasul cu partiționarea ai opțiuni de partiționare manuală (pentru experți) sau automată - adică installer-ul taie din felia Window-ului ca să dea spațiu pe hard disk lui Ubuntu. Mare atenție pentru începători la acest capitol ca să nu pierdeți tot ce aveți pe hard dintr-o neatenție ori lipsă de documentare (citiți documentația mai întâi, de pe situl ubuntu).

Nou și remarcabil! Poți instala Ubuntu și mai simplu, din Windows. Dacă nu te pricepi deloc, introduci CD-ul în timp ce rulezi Windows și îți apare o fereastră ce te invită să instalezi Ubuntu pe un spațiu liber de pe hard disk, fără să afecteze funcționarea Windows-ului. Metoda funcționează, fără afectarea sensibilă a performanțelor de rulare a sistemului de operare Ubuntu 8.10.

**After install.** Scoți CD-ul, îl pui la păstrare. Pornești computer-ul și te întâmpină un meniu de boot din care alegi sistemul de operare pe care vrei să intri (dacă ai mai multe). Spre deosebire de versiunile anterioare, Ubuntu 8.04 se încarcă neobișnuit de repede, în 10-15 secunde, acesta fiind real motiv de bucurie. Programele preinstalate, filme, muzică, YouTube și Mess...

**Să folosim Ubuntu.** Ce putem face cu el? Intrăm în meniul "Aplicații", accesibil din bara superioară în stânga. Avem acolo câteva submeniuuri, de fapt categorii de programe; în Ubuntu, aplicațiile nu se află toate la grămadă ca pe Windows. Vrei dicționar (englez ce-i drept), calculator, bloc notes sau utilitar pentru realizarea de screenshot-uri, apelezi la prima categorie, "Accesorii". La "Birou" găsim calendar, email și editoarele similare Microsoft Word, Excell și PowerPoint din cea mai nouă versiune 2.4 (lansată recent) a suitei OpenOffice. Legat de partea office/grafică, Ubuntu 8.10 dispune de un număr de fonturi impresionant față de versiunile anterioare. La "Grafică" ne întâmpină F-Spot – viewer, manager și editor de poze, o aplicație pentru scanere, un Paint îmbunătățit și Gimp în ultima sa versiune, vizibil mai deșteaptă și mai complexă. Gimp de altfel, este un Adobe Photoshop care nu costă 2.000 de dolari, ci e foarte gratis. "Jocuri" sunt peste 15, d'alea banale și nemuritoare.

Categoria "Internet" include Evolution (client e-mail), ultima versiune a Pidgin – multimessenger, un downloader pentru torrente, iar ca browser web Firefox 3. Categoria "Muzică și filme" vine cu "Brasero" pentru inscripționarea CD/DVD-urilor (Nero, pe înțelesul unora), utilitare pentru înregistrarea sunetului de pe microfon sau line-in și de conversie a CD-urilor Audio în formate digitale comprimate, player de filme și Rhythmbox – un player audio complex (cei care preferă Winamp, pot instala echivalentul Audacious).

Probleme legate de redarea filmelor, melodiilor: MP3, MPEG, flash, Real, QuickTime sunt considerate formate proprietare și de aceea nu merg din start. Dar e simplu. Dai clic pe un MP3 și îți va apărea o fereastră în care ți se solicită acordul de a descărca codecul necesar pe răspunderea ta. La fel se întâmplă cu filmele ori când intri prima dată pe YouTube și nu poți

vedea un clip video flash. Ca să se poată instala codecurile și plugin-urile necesare trebuie să fii conectat la net.

Pentru a instala alte aplicații, nu trebuie neapărat să pierzi vremea pe Google. Intri în meniul Sistem, clic pe Administrare, apoi pe “Administratorul de pachete Synaptic”. Interfața acestui program e simplă și intuitivă, te prinzi instant de modul de utilizare.

În concluzie, chiar și fără să mai cauți și instalezi software-uri suplimentare Ubuntu 8.10 se prezintă ca un sistem de operare perfect utilizabil imediat după instalarea sa, măcar pe partea de office, editare grafică și internet. Este mai stabil decât versiunile anterioare. A te conecta la rețele Wi-Fi este o nimica toată (un clic, eventual să bagi o parolă). Mici/mari neajunsuri, cum ar fi lipsa driverelor pentru unele TV tunere ori suportul restrâns pentru conexiunea video DVI încă are comparativ cu Windows – să nu uităm că majoritatea producătorilor de accesorii hardware sprijină doar sistemul de operare de la Microsoft din punct de vedere al driverelor.

## GLUME DE INFORMAT...

Un specialist în calculatoare și un specialist în automobile se contraziceau:

*Calculatoristul:* Eu dacă proiectam automobile, în 10 ani aveau viteza dublă și consumul de combustibil pe sfert!

*Automobilistul:* Dacă proiectai tu automobilul, când apăsați pe pedala de frână apărea pe un ecran întrebarea: Chiar doriți să frânați acum?

Programatoarea Florica caută pe net în timpul programului de lucru rețete de gătit. Google i-a dat 24.807 rezultate, Yahoo -18.399, iar MSN - 16.005.

Cu cât este Google mai dăunător decât MSN pentru procesul de muncă?

- Userul Gigel a decis să-și ia calculator. Un computer nou costă în magazine 1000 euro. Dar, deoarece Gigel avea deja blocul de alimentare, el a hotărât să-și ia pe componente și să-l asambleze acasă. A cumpărat placa de bază cu 150 euro, procesorul cu 200 euro, harddul cu 150 euro, și monitorul cu 300 euro.

Câți euro a economisit Gigel, dacă după 3 secunde de funcționare blocul de alimentare a ars celelalte componente.



# PREMII PROSOFT@NT



## ***Secțiunea programare***

### *Premiul I*

Budău Adrian, clasa a IX-a, Colegiul Național "Gh. Vrânceanu", Bacău  
Băieșu Vlad, clasa a X-a, Colegiul Național de Informatică, Piatra-Neamț  
Vicol Sergiu, clasa a X-a, Colegiul "C. Negruzzi", Iași  
Munteanu Daniel, clasa a XI-a, Colegiul Național de Informatică, Piatra-Neamț

### *Premiul II*

Paval Onisim clasa a IX-a, Colegiul Național " Petru Rareș ", SV  
Miluț Petronela, clasa a X-a, Colegiul Național "Roman Vodă", Roman  
Mihăilă Ștefan, clasa a XI-a, Colegiul Național de Informatică, Piatra-Neamț

### *Premiul III*

Popescu Dragoș, clasa a XI-a, Colegiul Național de Informatică, Piatra-Neamț

### *Mențiuni*

Atomei Cosmin, clasa a XI-a, Colegiul Național de Informatică, Piatra Neamț  
Ghiuță Alexandru, clasa a XI-a, Colegiul Național "Roman Vodă", Roman  
Ionescu Ștefan, clasa a XII-a, Colegiul Național de Informatică, Piatra-Neamț  
Marcu Florian, clasa a XI-a, Colegiul Național "Roman Vodă", Roman  
Paicu Alexandru, clasa a XII-a, Liceul de Informatica "Grigore Moisil", Iasi  
Petrov Robert, clasa a XII-a, Liceul de Informatica "Grigore Moisil", Iasi  
Racec Emil, clasa a XI-a, Colegiul Național "Gh. Vrânceanu ", Bacău  
Traistă Alexandru, clasa a XII-a, Colegiul Național "Ștefan Cel Mare", Suceava  
Asandei Radu, clasa a XII-a, Colegiul Național de Informatică, Piatra-Neamț

## ***Secțiunea soft educațional***

### *Premiul I și premiul special*

Porof Victor, Dima Teodor , Colegiul Național "E. Racoviță", Suceava

### *Premiul I*

Adoamnei Andrei ,Colegiul Național "Mihai Eminescu", Suceava

### *Premiul II*

Ciucă Alin Gabriel, Colegiul Național " Petru Rareș ", Piatra-Neamț

### *Premiul III*

Rotaru Mihai, Zait Madalina , Colegiul Național de Informatică, Piatra-Neamț

### *Mențiune*

Coman Alexandru, Colegiul Național "Ștefan Cel Mare", Suceava

## ***Secțiunea soft utilitar și pagini web***

### *Premiul I*

Budișteanu Ionuț, Colegiul Național "Mircea cel Batran", Vâlcea

### *Premiul II*

Buciu Alina , Lungoci Ioana Colegiul Național de Informatică, Piatra-Neamț

### *Premiul III*

Bîrgu Florin, Spataru Cosmin, Barna Victor, Colegiul Național "Ferdinand I", Bacău

### *Mențiune*

Tabacariu Ioan, Colegiul Național de Informatică, Piatra-Neamț

