

MODEL TEST 7

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila x memorează un număr natural de exact 6 cifre. Care dintre următoarele instrucțiuni C/C++ elimină din x cele două cifre situate la mijlocul numărului?

- a. $x/10000*100+x\%100$; b. $x=x/100-x/100\%100+x\%100$;
c. $x=x/10000+x\%100$; d. $x\%10000+x/100$;

2. Se consideră z o variabilă globală care are valoarea 5 și următorul subprogram f :

```
int z=5;
void f(int x, int &y)
{ x++; y+=x; z+=y; }
```

Care va fi valoarea variabilei z după apelul $f(z,z)$?

- a. 16 b. 22 c. 5 d. 24

3. Se consideră următoarele declarații:

```
struct Elev {char nume[30]; float mg;};
Elev E[100];
```

Care dintre următoarele variante reprezintă numele primului elev din vectorul E ?

- a. $E[0].nume$ b. $Elev[0].nume$ c. $E.nume[0]$ d. $Elev.E[0].nume$

4. Utilizând metoda backtracking, se generează toate secvențele formate din n caractere din mulțimea $\{'+', '-','o'\}$, astfel încât în orice prefix al unei secvențe astfel generate numărul de caractere '-' nu depășește numărul de caractere '+'. De exemplu, pentru $n=3$, s-au generat (în această ordine) următoarele secvențe:

+++ , ++o , ++- , +o+ , +oo , +o- , +-+ , +-o , o++ , o+o , o+- , oo+ , ooo.

Dacă utilizăm același algoritm pentru a genera secvențele de lungime $n=4$, care va fi cea de a opta secvență generată?

- a. +--o b. +++- c. +-+o d. ++++

5. Un graf neorientat are 50 de vârfuri și 5 componente conexe. Numărul maxim de muchii pe care le poate avea acest graf este:

- a. 225 b. 575 c. 2070 d. 1035

SUBIECTUL II

(40 puncte)

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[x]$ partea întreagă a numărului natural x .

- Scriveți valoarea afișată în urma executării algoritmului dacă se citesc numerele 1983 și 237. (6p.)
- Scriveți o pereche de numere care poate fi citită, astfel încât în urma executării algoritmului, să se afișeze valoarea 11. (6p.)
- Scriveți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scriveți în pseudocod un algoritm, echivalent cu cel dat, care să înlocuiască a doua structură repetitivă cu o structură repetitivă alt de alt tip. (6p.)

citește a, b (numere naturale) $x \leftarrow 0$

```

execută
  x ← x*10+a%10+b%10
  a ← [a/10]
  b ← [b/10]
cât timp a=0 sau b=0

```

 $n \leftarrow 0$

```

repetă
  n ← n*10+x/10%10
  x ← [x/100]
până când x=0
scrie n

```

2. În declararea alăturată, variabila x memorează, pentru fiecare dintre cei 30 de elevi dintr-o clasă, numărul matricol, precum și data nașterii elevului. Scrieți o secvență de instrucțiuni C/C++ care citește de la tastatură informațiile despre elevii din clasă și afișează pe ecran numărul matricol al ultimului elev care are anul nașterii număr par, sau mesajul **nu există** în caz contrar. (6p.)

```

struct elev
{ int nr_mat;
  struct
  { int zi,luna,an;
  }data;
}x[30];

```

3. În secvența de instrucțiuni de mai jos, variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 7 linii și 7 coloane, numerotate de la 1 la 7, având elemente de tip **char**. Fără a utiliza alte variabile decât cele menționate, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat. (6p.)

```

for(i=1;i<=7;i++)
  for(j=1;j<=7;j++)
    .....

    !####!
    ?!###!?
    ??#!??
    ???!???
    ??#!??
    ?!###!?
    !#####

```

SUBIECTUL III

(30 puncte)

1. Subprogramul **radical** are patru parametri:
- **n**, prin care primește un număr natural nenul ($n \leq 10^9$)
 - **k**, prin care primește un număr natural mai mare sau egal cu 2
 - **a**, prin care furnizează numărul maxim extras din radicalul de ordin **k**, al numărului **n**
 - **b**, prin care furnizează numărul minim ce rămâne sub radicalul de ordinal **k**.

Scrieți definiția completă a subprogramului.

Exemplu: dacă $n=2020$ și $k=2$, atunci $\sqrt[2]{2020} = 2\sqrt{505}$, adică **a** va fi **2**, iar **b** va fi **505**.

(10 p.)

2. Un text are cel mult 200 de caractere, iar cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe spații. Scrieți un program Pascal/C/C++ care citește de la tastatură textul și care codifică acest text prin transformarea literelor din cuvintele care au număr egal de vocale și consoane astfel: litera a va deveni litera z, litera b va deveni litera y, ... , litera z va deveni litera a. Textul final va avea aceeași lungime cu textul inițial, spațiile libere dintre cuvinte se păstrează.

Exemplu textul: **azi avem simulare la info**

Se va afișa textul: **azi zevn hrnfoziv oz rmul**

(10 p.)

3. Fișierul **cifre.in** conține pe prima linie cel mult 10^6 cifre. Să se citească cifrele din fișier și să se rearanjeze dacă este posibil astfel încât să formeze cel mai mare număr palindrom, sau mesajul **IMPOSIBIL**. Afișarea se va face pe ecran. Proiectați un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare.

Exemple:

Cifre.in	2 0 1 9 2 0 2 0	2 2 1 2 0 2 0
Pe ecran	IMPOSIBIL	2201022

- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2 p.)
- b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat. (8 p.)

REZOLVARE SUBIECTE
TEST 7
SUBIECTUL I
(20 de puncte)

1b	2b	3a	4c	5d	5x4p
----	----	----	----	----	------

SUBIECTUL al II – lea
(40 de puncte)
1.

- a) După execuția primei structuri repetitive vom obține $x=1121$
 După execuția celei de-a doua structuri repetitive vom obține $n=21$
 Răspuns **21**
- b) Orice pereche de numere care să furnizeze după prima structură repetitivă valoarea x de forma **a1b1** va da răspunsul corect. O astfel de pereche este: $a=2131$ și $b=5050$

c) Programul C/C++:

```
#include <iostream>
using namespace std;
int main()
{
    int a,b, x, n=0;
    cin>>a>>b;
    x=0;
    do{
        x=x*10+a%10+b%10;
        a=a/10;
        b=b/10;
    }while(a!=0&& b!=0);
    if(x!=0)
    do{
        n=n*10+x/10%10;
        x=x/100;
    }while( x!=0);
    cout<<n;
    return 0;}

```

d) Algoritmul în pseudocod echivalent care înlocuiește a doua structură repetitivă este:

```
citeste a,b (numere naturale)
x←0
execută
    x← x*10 + a%10+b%10
    a← [a/10]
    b← [b/10]
■cat timp a≠0 și b≠0
n←0
    cat timp x≠0 executa
        n←n*10+x/10%10
        x← [x/100]
    ■
scrie n

```

2. Secvența de program care rezolvă cerința:

```
int poz = -1; // poz = poziția ultimului copil al cărui an de naștere este număr par
for( int i=0 ; i<30 ; i++)
{
    cin>>x[i].nr_mat>>x[i].data.zi>>x[i].data.luna>>x[i].data.an;
    if( x[i].data.an %2 ==0) poz = i;
}
if(poz == -1) cout<<"nu exista";
else cout<<x[poz].data.nr_mat;

```

3. Instrucțiunile care ar putea fi scrise în locul punctelor de suspensie pentru a construi matricea din exemplu, sunt:

```
if( i==j || i+j==8) a[i][j]='!';
else if( i<j && i+j<8 && i<4 || i>j && i+j>8 && i>4) a[i][j]='#';
    else a[i][j]='?';
```

SUBIECTUL al III – lea
(40 de puncte)

1. void radical(int n,int k, int &a, int &b)

```
{
    ///descompunem in factori primi
    a=1; b=1;
    int e,i,d=2;
    while(n>1)
    {
        e=0;
        while(n%d==0)
        {
            e++;
            n=n/d;
        }
        /// modificarea lui a
        for(i=1;i<=e/k;i++)
            a=a*d;
        /// modificarea lui b
        for(i=1;i<=e%k;i++)
            b=b*d;
            d++;
        }
    }
```

2. #include<iostream>

```
#include<cstring>
using namespace std;
char s[202];
int i,p1,p2,l,j,ok;

int main()
{ cin.get(s,201,'\n');
  l=strlen(s);
  for(i=0;i<l;i++)
  {
    ///delimitam cuvintele- p1 este indicele de inceput al cuvântului utilizat din text
    if ((i==0 or s[i-1]==' ') and s[i]!=' ') p1=i;
    ///p2- este indicele de finalul al cuvântului
    if ((i==l-1 or s[i+1]==' ') and s[i]!=' ')
    {
      p2=i;
      ///parcurem caracterele din cuvânt de la indicii situați în intervalul [p1,p2] și
      ///calculăm numărul de vocale
      ok=0;
      for(j=p1;j<=p2;j++)
        if (strchr("aeiou",s[j])!=NULL) ok++;
      if (ok*2==p2-p1+1)
```

```

    {
        //dacă dublul numărului de vocale este egal cu lungimea cuvântului, facem
        // modificarea literelor
        for(j=p1;j<=p2;j++)
            s[j]='a'+(25- (s[j]-'a'));
    }
}
}
cout<<s; }

```

3. a. Fiind doar cifre în fișierul de intrare, vom defini un vector de frecvență de dimensiune 10. Pentru fiecare cifră vom stabili numărul ei de apariții. Problema nu va avea soluție dacă există 2 sau mai multe cifre cu frecvențe impare. Dacă există soluție, cifrele se vor afișa în două seturi, folosind în fiecare set un număr de cifre egal cu jumătate din frecvența fiecărei cifre. Palindromul de valoare maximă va începe cu un set de cifre alese în sens descrescător, va continua cu cifra cu frecvență impară (dacă există) și se va termina cu al doilea set de cifre alese în sens crescător.

b. **#include<fstream>**

#include<iostream>

using namespace std;

ifstream fin("cifre.in");

int x,f[10],c,imp,i;

int main()

{

while(fin>>x)

f[x]++;

imp=0;

for(c=0;c<=9;c++)

if (f[c]%2==1) imp++;

if (imp>1) cout<<"IMPOSIBIL";

else

{

///af în ordine descrescătoare cifrele, fiecare fiind scrisă de jumătate din frecvența sa

for(c=9;c>=0;c--)

for(i=1;i<=f[c]/2;i++)

cout<<c;

///verificam daca avem o frecvență impară

for(c=0;c<=9;c++)

if (f[c]%2==1) cout<<c;

///af în ordine crescătoare cifrele, fiecare fiind scrisă de jumătate din frecvența sa

for(c=0;c<=9;c++)

for(i=1;i<=f[c]/2;i++)

cout<<c;

}

}

BAREM DE EVALUARE ȘI DE NOTARE**TEST 7**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 puncte)**

Răspuns	Punctaj
1b 2b 3a 4c 5d	5x4p.

SUBIECTUL II**(40 puncte)**

1.	a) Răspuns corect: 21	6p.	
	b) Răspuns corect: orice pereche de numere care furnizează rezultatul final 11	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are a doua structură repetitivă conform cerinței, principal corectă, dar nu este echivalent cu cel dat.
2.	Pentru rezolvare corectă - acces la câmpurile de pe primul nivel al înregistrării - acces la câmpurile de pe al doilea nivel al înregistrării - afișare conform condiției impuse(*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (condiție corectă, afișare pentru fiecare caz) conform cerinței.
3.	Pentru rezolvare corectă - acces la un element al tabloului - atribuire a valorilor indicate elementelor tabloului (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru atribuirea valorilor conform cerinței, pentru elementele situate între cele două diagonale și 2p. pentru atribuirea valorilor elementelor situate pe diagonala secundară și pe diagonala principală.

1) Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL III**(30 puncte)**

1	Pentru subprogram corect - antet subprogram (*) - determinarea numerelor cerute (**) - declarare a tuturor variabilelor locale, corectitudine globală a subprogramului)	10p. 3p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametrii de intrare, parametrii de ieșire) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (factori primi și exponenți + modificare variabila a + modificare variabila b)
2	Pentru program corect - declarare a unei variabile care să memoreze un șir de caractere - citirea textului - modificarea textului conform cerinței (*) - afișarea textului - declarare a variabilelor simple, - corectitudine globală a programului)	10p. 1p. 1p. 5p. 1p. 1p. 1p	(*) Se acordă câte 1p. pentru fiecare aspect - obținerea unui cuvânt - determinarea lungimii unui cuvânt, - determinarea nr de vocale - determinarea nr de consoane - codificarea unui cuvânt care respecta cerința
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului)	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă este folosirea vectorilor de frecvență. Cazul IMPOSIBIL , e îndeplinit dacă avem 2 sau mai multe cifre cu frecvențe impare. Soluția va fi obținută prin afișarea descrescătoare a cifrelor și afișarea crescătoare a cifrelor, în fiecare parte punând jumătate din câte am citit. Caz special ce cifra va fi la mijloc, dacă avem 1 singură frecvență impară?