

MODEL TEST 4

- 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.

Subiectul I**(20 de 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. Precizați care din expresiile de mai jos este adevărată pentru **a** cel mai mare număr natural de patru cifre distincte și **b** cel mai mic număr natural de patru cifre distincte.

- a) $(a/b == 8) \parallel (a \% b == 0)$
b) $(a/b == 9) \&\& (a \% b > 0)$
c) $(a \% b == 8) \parallel (a/b == 0)$
d) $(a \% b == 9) \&\& (a/b > 0)$

2. Fie tabloul unidimensional (20, 15, 12, 8, 4, 2, 1). Pentru a verifica dacă numărul 5 se află printre elementele tabloului, se aplică metoda căutării binare. Care este succesiunea corectă de elemente cu care se compară numărul căutat?

- a. 1,2,4,8
b. 8,4,2
c. 1,2,4,8,12,15,20
d. 8,2,4

3. Determinați numărul grafurilor parțiale distincte cu număr impar de arce pentru graful orientat dat prin matricea de adiacență următoare:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- a. 8
b. 128
c. 120
d. 5

4. Fie subprogramul f definit alăturat. În urma executării secvenței

```
int a=0; f(a);  
se vor afișa valorile:
```

- a. 321
b. 123
c. 000
d. 333

```
void f(int &x)  
{ x++;  
if (x<3)f(x);  
cout<<x;}
```

5. Un program folosește un algoritm de tip backtracking și generează, în ordine lexicografică, toate anagramele distincte ale cuvântului **irisi**. Primele 5 anagramme generate de acest algoritm sunt **iiirs**, **iiisr**, **iiris**, **iirsi**, **iisir**. Care este cea de a zecea anagramă generată de acest program?

- a. irsii
b. isiri
c. riiis
d. isiiir

Subiectul II**(40 de puncte)**

1. În algoritmul următor 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 real x :

citește a,b(numere naturale)

p ← 1

cât timp $a * b > 0$ și $a \% 10 = b \% 10$ execută

a ← [a/10]

b ← [b/10]

p ← p * 10

a ← a * p

scrie a

- a. Scrieți valoarea afișată dacă se citesc valorile 51467 și 67. (6p)
- b. Dacă pentru **b** se citește valoarea 73, scrieți numărul valorilor de 4 cifre ce pot fi citite pentru a astfel încât valoarea afișată să fie divizibilă cu 50. (6p)
- c. Scrieți un algoritm echivalent cu cel dat care să utilizeze o structură repetitivă cu test final. (6p)
- d. Scrieți programul C/C++ corespunzător algoritmului dat (10p)

2. În secvența de instrucțiuni de mai jos, variabilele *i* și *j* sunt de tip întreg, iar variabila **A** este de tip tablou bidimensional cu 5 linii și 5 coloane cu numere întregi. Fără a utiliza alte variabile, înlocuiți cu o instrucțiune punctele de suspensie din secvența de mai jos astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă conținutul alăturat:

```
for (i=1;i<=5;i++)
    for (j=1;j<=5;j++)
        .....
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 0 \\ 2 & 4 & 1 & 3 & 0 \\ 3 & 1 & 4 & 2 & 0 \\ 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(6p)

3. În declarația alăturată, variabila **A** memorează în câmpurile **x** și **y** coordonatele carteziene ale unui punct din planul **xOy**, iar variabila **C** memorează raza și coordonatele centrului unui cerc.

```
struct punct
    {float x,y;} A;
struct cerc
    {punct p; float r;} C;
float d;
```

Scrieți o secvență de instrucțiuni prin care se calculează în variabila **d** distanța de la punctul **A** la centrul cercului și apoi se verifică poziția punctului față de cerc afișându-se, în funcție de rezultatul obținut, mesajul INTERIOR, EXTERIOR, respectiv PE CERC.

(6p)

Subiectul III

(30 de puncte)

1. Subprogramul **nrmax** primește prin parametrii **a** și **b** două numere naturale cu cel mult 9 cifre fiecare și returnează prin parametrul **c** cel mai mare număr cu cifrele impare distincte care apar atât în **a** cât și în **b** sau 0 dacă nu există astfel de cifre.

Scrieți definiția completă a subprogramului.

Exemplu: dacă $a=45013$ și $b=215578$ atunci se obține $c=51$.

(10 p)

2. Se consideră un text cu maxim 250 de caractere (litere mici și spații) în care cuvintele sunt separate printr-un singur spațiu. Scrieți un program care citește un astfel de text și apoi construiește șirul obținut prin modificarea tuturor cuvintelor de lungime pară prin interschimbarea între ele a celor două jumătăți ale cuvântului, păstrând ordinea caracterelor din cele două jumătăți. Programul afișează pe ecran textul astfel modificat, iar în cazul în care nu sunt cuvinte modificate se va afișa mesajul **TEXT NEMODIFICAT**.

Exemple: pentru textul **totul va fi bine** se va obține textul **totul av if nebi**, iar pentru textul **mama tata** se va afișa mesajul **TEXT NEMODIFICAT**

(10 p)

3. Fișierul **BAC.IN** conține cel mult un milion de numere naturale de cel mult 9 cifre fiecare separate prin câte un spațiu. Să se afișeze pe ecran cel mai mare număr **a** de exact 3 cifre care nu apare în fișier pentru care există un număr **b**, care apare în fișier, astfel încât **a+b=1000**.

Dacă nu există niciun astfel de număr se va afișa mesajul **NU EXISTA**.

Proiectați un algoritm eficient din punct de vedere al timpului de executare și justificați eficiența acestuia.

Exemple:

BAC.IN	22 8425 4567 6 8 999 33 995 800 200 45123	8425 994 4567 5 6 994 3300 995 800 200 45123
Pe ecran	994	NU EXISTA

(10 p)

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

Subiectul I

1. b	2. d	3. b	4. d	5. d
------	------	------	------	------

Subiectul II

1. a. 51400

b. 450 de valori (toate numerele de forma $\overline{xy73}$, $\overline{xy03}$, $\overline{xy53}$, $\overline{xy50}$, $\overline{xy00}$, $x \in \{1,2,\dots, 9\}$, $y \in \{0,1,\dots, 9\}$)

c. citește a,b (numere naturale)
 $p \leftarrow 1$
 dacă $a*b > 0$ și $a \% 10 = b \% 10$ atunci
 repetă
 $a \leftarrow [a/10]$
 $b \leftarrow [b/10]$
 $p \leftarrow p*10$
 până când $a*b = 0$ sau $a \% 10 \neq b \% 10$
 $a \leftarrow a*p$
 scrie a

d.

```
#include<iostream>
using namespace std;
int main()
{ int a,b,p=1;
cin>>a>>b;
while (a*b>0 && a%10==b%10)
{ a=a/10;
  b=b/10;
  p=p*10;
}
a=a*p;
cout<<a;
return 0;
}
```

2. $A[i][j] = (i*j) \% 5$;

```
3. d=sqrt((A.x-C.p.x)*(A.x-C.p.x) + (A.y-C.p.y)*(A.y-C.p.y));
if (d<C.r) cout<<"INTERIOR";
else if (d>C.r) cout<<"EXTERIOR";
else cout<<"PE CERC";
```

Subiectul III

```
1. void nrmax (long a, long b, long &c)
{ int i, ok1, ok2; long a1, b1;
c=0;
for (i=9;i>=1;i-=2)
{ ok1=ok2=0; a1=a; b1=b;
while (a1>0&&ok1==0) if (a1%10==i)ok1=1;
else a1/=10;
while (b1>0&&ok2==0) if (b1%10==i) ok2=1;
else b1/=10;
if (ok1==1 && ok2==1) c=c*10+i;
}
}
```

```

2. #include<iostream>
   #include<cstring>
   using namespace std;
   int main()
   { char s[251], t[ ], c[ ], *p, z[ 251];
     int ok=0;
     cin.get (s, 251);
     z[0]=NULL;
     p=strtok(s," ");
     while (p) { if (strlen(p)%2==0){
                   strcpy(c,p);
                   strcpy(t, p+strlen(p)/2);
                   p[strlen(p)/2]=NULL;
                   strcat (t,p);
                   if (strcmp (c,t)) {
                       ok=1;
                       strcpy(p,t);
                   }
                   }
                   strcat(z,p);
                   strcat(z," ");
                   p=strtok(NULL," ");
           }
     if (ok) cout <<z;
     else cout << "TEXT NEMODIFICAT";
     return 0;
   }
3. #include<iostream>
   #include<fstream>
   using namespace std;
   int v[1000];
   ifstream f ("BAC.IN");
   int main()
   {long x, ok=0;
     while (f>>x) if (x<1000)v[x]++;
     x=999;
     while (x>=100&&ok==0)
         if (v[x]==0 && v[1000-x]!=0) ok=1;
         else x--;
     if (ok) cout<<x;
     else cout<<"NU EXISTA";
     return 0;
   }

```

O soluție posibilă folosește un vector de frecvență pentru numerele de maxim 3 cifre care apar în fișier. Apoi se parcurge vectorul de la ultima poziție și se identifica primul număr (poziție) care are frecvența nulă și pentru care se îndeplinește condiția că $v[1000-nr]$ este nenulă. Complexitatea algoritmului este liniară ($O(n)$).

BAREM DE EVALUARE ȘI DE NOTARE

TEST 4

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

Subiectul I

20 de puncte

1. b	2. d	3. b	4. d	5. d	4p×5
------	------	------	------	------	------

Subiectul II

40 de puncte

1. a. 51400	6p
b. 450 de valori (toate numerele de forma $\overline{xy73}$, $\overline{xy03}$, $\overline{xy53}$, $\overline{xy50}$, $\overline{xy00}$)	6p
c. echivalare corectă	6p
d. Pentru program corect	10p
-declararea corectă a tuturor variabilelor	2p
-citire corectă	1p
-scriere corectă	1p
-instrucțiune repetitivă corectă	3p
-atribuiri corecte	2p
-corectitudinea globală a programului	1p
2. $A[i][j]=(i*j)\%5;$	6p
3. Pentru rezolvare corectă	6p
-accesul corect la câmpurile de pe primul nivel al înregistrării	1p
-accesul corect la câmpurile de pe al doilea nivel al înregistrării	1p
-determinarea corectă a valorii variabilei d	2p
-instrucțiunea de decizie care afișează corect mesajul cerut	2p

Subiectul III

30 de puncte

1. Pentru subprogram corect	10 p
-structură antet corect	2p
-declarare corectă a parametrilor	1p
-determinare cifre distincte comune	3p
-determinare număr maxim cerut	3p
-declarare a tuturor variabilelor locale, corectitudine globală a subprogramului	1p
2. Pentru rezolvare corectă	10p
-declarare corectă a unei variabile care să memoreze un șir de caractere	1p
-citirea șirului de caractere	1p
-separarea cuvintelor din frază	2p
-modificarea corectă a cuvintelor cu lungimea pară	2p
-construirea corectă a șirului cerut	2p
-afișarea mesajului	1p
-corectitudine globală a programului	1p
3. Pentru rezolvare corectă	10p
- operații cu fișiere (declarare, deschidere)	1p
- citirea numerelor	1p
- algoritm principial corect	2p
- determinarea valorii cerute	2p
- afișarea rezultatului	1p
- corectitudine formală (declarare variabile, structură program, sintaxa instrucțiunilor)	1p
- coerența explicării metodei	1p

<p>- explicarea unor elemente de eficiență conform cerinței (numai pentru eficiență $O(n)$)</p> <p>O soluție posibilă folosește un vector de frecvență pentru numerele de maxim 3 cifre care apar în fișier. Apoi se parcurge vectorul de la ultima poziție și se identifică primul număr (poziție) care are frecvența nulă și pentru care se îndeplinește condiția că $v[1000-nr]$ este nenulă. Astfel, algoritmul utilizat este de tip liniar (complexitate $O(n)$).</p>	1p
---	----