MODEL TEST 8

- *Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.*
- *Timpul de lucru efectiv este de 3 ore.*
- *Identificatorii utilizaţi în rezolvări trebuie să respecte precizările din enunţ (bold), iar în lipsa unor precizări explicite, notaţiile trebuie să corespundă cu semnificaţiile 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, scrieţi pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila **n** este de tip întreg. Care dintre următoarele expresii **C/C++** are valoarea **1** dacă și numai dacă valoarea memorată de **n** este divizibilă cu **3** și nenulă?

a) **n%10==3** **b)** **n/3==0** **c)** **(n-3)%3==0** **d)** **n%2==1**

2. Funcția **f** are definiția alăturată. Ce se va afișa la apelul **f(4)**?

```
void f(int n)
{  int i;
   for(i=1;i<=n;i++)
   {  if(i%2==1)f(i-1);
      cout<<i;
   }
}
```

a) **1214** **b)** **121234** **c)** **112234** **d)** **123412**

3. Utilizând metoda backtracking se generează toate șirurile de **4** valori din mulțimea {**1,2,3,4,5**} astfel încât pe oricare două poziții alăturate să nu se afle două valori de aceeași paritate. Primele **7** soluții generate sunt: {**1,2,1,2**}, {**1,2,1,4**}, {**1,2,3,2**}, {**1,2,3,4**}, {**1,2,5,2**}, {**1,2,5,4**}, {**1,4,1,2**}. Care este a **8**-a soluție?

a) **{2,1,3,1}** **b)** **{1,4,3,2}** **c)** **{2,3,2,1}** **d)** **{1,4,1,4}**

4. Se consideră un graf cu **11** vârfuri și **55** de muchii. Numărul minim de muchii care trebuie eliminate din graf astfel încât să se obțină un graf eulerian este:

a) **10** **b)** **11** **c)** **0** **d)** **1**

5. Un graf orientat are **7** noduri numerotate de la **1** la **7** și arcele: **(1,2), (2,3), (4,5), (5,6), (6,4), (7,5)**. Care este numărul minim de arce care trebuie adăugate astfel încât graful să devină tare conex?

a) **1** **b)2** **c) 3** **d) 0**

Subiectul II (40 puncte)

Scrieţi pe foaia de examen răspunsul pentru fiecare dintre cerinţele următoare.

1. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu **x%y** restul împărţirii numărului natural **x** la numărul natural nenul **y** şi cu **[z]** partea întreagă a numărului real **z**.

a) Scrieți ce valoare se va afișa, dacă se citesc valorile **895** și **124**. **(6p.)**

b) Scrieți toate perechile de numere ce pot fi citite pentru **a** și **b**, cu **a < b**, astfel încât să se afișeze valoarea **2020**. **(6p.)**

c) Scrieți programul **C/C++** corespunzător algoritmului dat. **(10p.)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, care să nu conțină nicio structură repetitivă. **(6p.)**

```
citește a și b (numere naturale nenule, a și b <10^5)
p=1;e=0;
cat timp b≠0 executa
    p1=1;x=b%10;t=0;c=0;aux=a;
    cat timp aux≠0 executa
      y=aux%10
      d=y*x+t
      t=[d/10]
      d=d%10
      c=c+d*p1;
      p1=p1*10;
      aux=[aux/10];
    daca t≠0 atunci
                c=c+t*p1;
    e=e+c*p; p=p*10; b=[b/10]
scrie e;
```

2. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu **6** linii și **6** coloane, numerotate de la **0** la **5**, având inițial toate elementele egale cu valoarea -1. Fără a utiliza alte variabile, completați secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **a** să memoreze tabloul alăturat.**(6p)**

```
for(i=0;i<6;i++)
      for(j=0;j<6;j++)
      ...................
```

5	1	2	3	4	0
1	4	1	2	0	4
2	1	3	0	2	3
3	2	0	2	1	2
4	0	2	1	1	1
0	4	3	2	1	0

3. Variabila **s** memorează simultan următoarele date despre fiecare dintre cele **40** de clase din cadrul unei școli: un cod de maximum **5** caractere, reprezentând numele clasei, numărul de elevi din clasă și mediile acestora. În fiecare clasă sunt maximum **30** de elevi. Știind că expresiile **C/C++** de mai jos au ca valori un șir de caractere ce reprezintă codul celei de a **2**-a clase, o valoare naturală ce reprezintă numărul de elevi din cea de a **2**-a clasă, respectiv o valoare reală ce reprezintă media celui de al **3**-lea elev din a **2**-a clasă, scrieți definiția unei structuri cu eticheta **clasa**, care permite memorarea datelor despre o clasă, și declarați corespunzător variabila **s**. **(6p.)**

s[1].cod s[1].NrElevi s[1].Medie[2]

SUBIECTUL III (30 puncte)

1. Subprogramul **inserare** are doi parametri:
- **n**, prin care primește un număr natural (**$n \in [10,10^5]$**);
- **d**, prin care furnizează numărul obținut prin inserarea între două cifre de aceeaşi paritate lui **n,** media lor aritmetică sau **-1** dacă acesta nu conţine două cifre de aceeaşi paritate alăturate.

Scrieți definiția completă a subprogramului.
Exemplu: dacă **n=1976**, după apel **d=159876,**iar pentru **n=1234**, după apel **d=-1**. **(10p.)**

2. Un text are cel mult **255** de caractere, iar cuvintele sale sunt formate numai din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program **C/C++** care citește de la tastatură un text de tipul precizat mai sus, și afișează pe ecran şirul modificat prin eliminarea din textul iniţial a acelor cuvinte în care vocalele sunt ordonate strict lexicografic. Un cuvânt format doar din consoane va fi eliminat din text. Dacă textul iniţial nu conţine cuvinte care îndeplinesc condiţia cerută se va afişa pe ecran doar mesajul **nu exista**. Se consideră vocale literele **a, e, i, o, u.**
Exemplu: pentru textul **ei aduc multe carti**, se va afişa textul **multe**, iar pentru textul **ea aduce multa bucurie** se va fişa mesajul **nu exista.** **(10p.)**

3. Șirul **f** este definit astfel: **$f_1=1$**, **$f_2=4$, $f_i=2*(f_{i-1}+1)-f_{i-2}$, $i \geq 3$.** Fişierul **bac.txt** conţine pe prima linie un număr natural **n ($1 \leq n \leq 10^9$)**, iar pe a doua linie cel mult **10^9** numere naturale din intervalul **$[1,100^2]$**. Se cere să se afișeze pe ecran numărul care ar apărea pe poziția **n** în șirul ordonat crescător obținut din toate numerele aflate pe a doua linie a fișierului care fac parte din șirul **f**. Dacă pe ultima linie a fișierului sunt mai puțin de **n** termeni ai șirului menționat anterior se afișează pe ecran mesajul **Nu exista**. Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fişierul **bac.txt** conţine numerele:
7
4 1 34 4 2 3 81 121 5 7 8 1 1 12 169 15 9 1 24
atunci se va afişa pe ecran valoarea **9**.
a) Descrieţi în limbaj natural algoritmul utilizat, justificând eficienţa acestuia. **(2p.)**
b) Scrieţi programul **C/C++** corespunzător algoritmului descris. **(8p.)**

REZOLVARE SUBIECTE TEST 8

Subiectul I

1. **c**
2. **b**
3. **d**
4. **c**
5. **b**

Subiectul II

1.

a) 110980

b) Perechile cerute sunt: 1 și 2020, 2 și 1010, 4 și 505, 5 și 404, 10 și 202, 20 și 101

c) Programul C/C++

```
#include <iostream>
using namespace std;
int main()
{
   int a,b,c,d,e,p,p1,aux,t,x,y;
   cin>>a>>b;
   p=1;e=0;
   while(b!=0)
   {
      p1=1;x=b%10;aux=a;t=0;c=0;
      while(aux!=0)
      {
         y=aux%10;
         d=y*x+t;
         t=d/10;
         d=d%10;
         c=c+d*p1;
         p1=p1*10;
         aux=aux/10;
       }
       if(t!=0)c=c+t*p1;
       e=e+c*p;
       p=p*10;
       b=b/10;
   }
   cout<<e;
   return 0;
}
```

d) Algoritmul calculează produsul celor două numere citite.

citește a și b (numere naturale nenule, a și b <10^5)
scrie a*b.

2.

```
for(i=0;i<6;i++)
  for(j=0;j<6;j++)
      if(i==j)a[i][j]=5-i;
      else if(i+j==5)a[i][j]=0;
          else if(j>i)a[i][j]=j-i;
              else a[i][j]=i-j;
```

3.

```
struct clasa
{
   char cod[5];
   int NrElevi;
   float Medie[30];
}s[40];
```

Subiectul III

1. Comparăm câte două cifre alăturate din număr.

```
void inserare(int n, int &d)
{
   int c1,c2,p,n1;
   n1=n;p=10;
   c1=n%10;d=c1;
   n=n/10;
   while(n!=0)
   {
      c2=n%10;
      n=n/10;
      if(c1%2==c2%2)
      {
         d=d+(c1+c2)/2*p;
         p=p*10;
      }
      d=d+c2*p;
      p=p*10;
      c1=c2;
   }
   if(d==n1)d=-1;
}
```

2. Extragem pe rând câte un cuvânt din text și verificăm dacă vocalele sunt ordonate strict crescător. În caz afirmativ îl eliminăm din șir. O posibilă soluție este următoarea.

```
#include <iostream>
#include <string.h>
using namespace std;

int main()
{
   char s[255],voc[]="aeiou",*p,*q,cuv[100];
   int i,poz,ok,gasit=0;
   cin.get(s,255);
   p=s;q=strchr(s,' ');
   while(q!=NULL)
```

```
        {
          cuv[0]=0;
          strncat(cuv,p,q-p);
          poz=-1;ok=1;
          for(i=0;i<strlen(cuv);i++)
            if(strchr(voc,cuv[i])!=NULL)
              if(poz==-1)poz=i;
              else if(cuv[i]<=cuv[poz])ok=0;
                  else poz=i;
          if(ok==1)
          {
            strcpy(p,q);gasit=1;p=p+1;
          }
          else p=q+1;
          q=strchr(p,' ');
        }
        poz=-1;ok=1;
        for(i=0;i<strlen(p);i++)
          if(strchr(voc,p[i])!=NULL)
            if(poz==-1)poz=i;
            else if(p[i]<=p[poz])ok=0;
                 else poz=i;
        if(ok==1){*p=NULL;gasit=1;}
        if(gasit==1)cout<<s;
        else cout<<"nu exista";
        return 0;
      }
```

3.

a) Se observă faptul că valorile din șir sunt pătrate perfecte. O soluție eficientă utilizează un vector de frecvență cu 100 elemente cu ajutorul căruia vom memora de câte ori a apărut valoarea k. Eficiența timp este dată de faptul că este un algoritm liniar, la o singura trecere prin fișier determinăm fiecare valoare din șir de câte ori apare; în funcție de numărul de apariții a fiecărei valori vom identifica valoarea cerută. Complexitatea algoritmului depinde doar de numărul de valori din fișier.

b) O posibilă soluție este următoarea:

```
#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;
ifstream fin("bac.txt");
int ap[101];
int main()
{
  int k,n,x;
  fin>>n;
  while(fin>>k)
    {
      x=floor(sqrt(k));
      if(x*x==k)ap[x]++;
    }
x=1;
while(x<=100&&n>0)
{
  n=n-ap[x];
  x++;
}
if(n<=0)cout<<(x-1)*(x-1);
else cout<<"nu exista";
return 0;
                    }
```

BAREM DE EVALUARE ȘI DE NOTARE TEST 8

- *Se punctează oricare alte modalităţi de rezolvare corectă a cerinţelor.*
- *Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracţiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărţirea punctajului total acordat pentru lucrare la 10.*
- *Utilizarea unui tip de date care depăşeşte domeniul de valori precizat în enunţ este acceptată dacă acest lucru nu afectează corectitudinea în funcţionarea programului.*

SUBIECTUL I (20 de puncte)

1. c 2. b 3. d 4. c 5. b	**5x4p.**

SUBIECTUL al II - lea (40 de puncte)

1.	**a)** Răspuns corect: 110980	**6p.**	
	b) Pentru răspuns corect	**6p.**	Se acordă câte un punct pentru fiecare dintre perechile: 1 și 2020, 2 și 1010, 4 și 505, 5 și 404, 10 și 202, 20 și 101
	c) Pentru program corect - declarare variabile - citire date - afişare date - instrucţiune de decizie - instrucţiuni repetitive (*) - atribuiri - corectitudine globală a programului[1)]	**10p.** 1p. 1p. 1p. 1p. 4p. 1p. 1p.	(*) Se acordă numai 2p dacă doar una dintre intrucțiunile repetitive este conform cerinței
	d) Pentru algoritm pseudocod corect - echivalenţă a prelucrării realizate, conform cerinţei (*) - corectitudine globală a algoritmului[1)]	**6p.** 5p. 1p.	 citește a,b scrie a*b
2.	**Pentru rezolvare corectă***	**6p.**	(*) Se acordă câte 1p. pentru fiecare aspect: (determinare valori de pe diagonala principala, diagonala secundară, zona de nord a matricei, zona est, zona sud, zona vest)
3.	**Pentru rezolvare corectă**	**6p.**	(*) Se acordă câte 2 puncte pentru fiecare aspect al cerinței (declare corectă pentru fiecare câmp al structurii)

SUBIECTUL al III - lea (30 de puncte)

1.	**Pentru subprogram corect** - antet subprogram (*) - determinare a numărului cerut (**) - declarare a variabilelor locale - corectitudine globală asubprogramului[1)]	**10p.** 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (testarea parității a două cifre alăturate, calcularea mediei aritmetice, inserarea unei cifre în număr).
2.	**Pentru program corect**	**10p**	(*) Se acordă câte 2p pentru fiecare aspect al

	- declarare a unei variabile care să memoreze un șir de caractere - citire a datelor - determinare șirului cerut (*) - afișare a datelor -declarare a variabilelor simple, corectitudine globală a programului[1)]	1p 1p 6p 1p 1p	cerinței(identificare cuvânt, verificare poziționare vocale, eliminare cuvânt din text)
3.	**a) Pentru răspuns corect** - coerenţa descrierii algoritmului (*) - justificare a unor elemente de eficienţă	**2p.** 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient.
	Pentru program corect - operaţii cu fişiere: declarare, pregătire în vederea citirii, citire din fişier - determinare a valorii cerute (*),(**) - utilizarea unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor, corectitudine globală a programului[1)]	**8p.** 1p. 5p. 1p. 1p.	(**) 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 Se observă faptul că valorile din șir sunt pătrate perfecte. O soluție eficientă utilizează un vector de frecvență cu 100 elemente cu ajutorul căruia vom memora de câte ori a aparut valoarea k. Eficiența timp este dată de faptul că este un algoritm liniar, la o singura trecere prin fișier determinăm numărul de apariții a fiecăre valori din șir, complexitatea algoritmului depinde doar de numărul de valori din fișier.

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