MODEL TEST 9

- *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 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. Dacă variabilele a și b sunt de tip int, ce valori vor avea variabilele a și b la finalul executării secvenței de instrucțiuni alăturate:

```
a=5; b=0;
do{
a--; b+=a*a;
} while(a>0);
```

a. a=0 și b=50
b. a= -1 și b=30
c. a=-1 și b=31
d. a=0 și b=30

2. Se consideră subprogramul cu definiţia alăturată. Ce valoare se va afişa în urma executării instrucţiunii de mai jos?
cout<<f(10); | printf("%d",f(10));

```
int f (int n)
{ int c;
if (n!=0)
 {if (n%2==1)
      c=1+f(n/2);
  else c=f(n/2);
  cout<<n%2; | printf("%d",n%2);
 return c;
}
else return 0; }
```

a. 01010 b. 1010 c. 10102 d. 21010

3. Un număr este palindrom dacă citit de la stânga la dreapta se obține același număr. Generând palindroamele de lungime 3, folosind cifrele 0, 1, 2, 3, 4 se obțin în ordine numerele: 101, 111, 121, 131, 141, 202, 212, …. Folosind același algoritm pentru a genera toate palindoramele pare de lungime 4 folosind cifrele 0, 1, 2, 3, 4, 5 care este al șaptelea număr generat?

a. 2002 b. 2442 c. 4004 d.4224

4. Care este numărul grafurilor orientate cu n noduri cu proprietatea că pentru orice pereche de noduri distincte i și j există cel puțin un arc între i și j.

a. 3^n b. n^3 c. $3^{n*(n-1)}$ d. $3^{n*(n-1)/2}$

5. Se consideră un arbore cu 8 noduri. Lista alăturată reține pentru fiecare nod al arborelui fiii lui (descendenții direcți). Câte dintre nodurile lui ar putea fi alese ca rădăcină astfel încât arborele să aibă număr maxim de niveluri?

1: -
2: -
3: 2, 6, 8
4: -
5: 1
6: 4, 7
7: 5
8: -

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

SUBIECTUL al II-lea (40 puncte)

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu **a%b** restul împărţirii numărului natural **a** la numărul natural nenul **b.**

a. Ce se va afișa dacă se citesc pentru n valoarea 5 și pentru x valorile: 16, 80, 48, 20, 240 **(6p)**

b. Dacă n=4, dați exemplu de patru valori pentru x pentru care algoritmul să afișeze 2020. **(6p)**

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

d. Scrieți un algoritm în pseudocod echivalent cu algoritmul dat în care să se utilizeze doar structuri repetitive condiționate posterior(cu test final) **(6p)**

```
citește n (număr natural)
d←0
pentru i←1,n execută
|       citește x (număr natural nenul)
|       dacă d=0 atunci
|       |       d←x
|       | altfel
|       |       repetă
|       |       |       r←x % d
|       |       |       x←d
|       |       |       d←r
|       |       |_ până când r=0
|       |       d←x
|       |_■
|_■
Scrie d
```

2. Variabila **p** memorează simultan informații referitoare la cei 100 angajați ai unei companii: numărul de identificare (un număr natural), numele (un șir cu maxim 50 caractere), salariul (un număr real), data nasterii și data angajării (ziua, luna și anul numere naturale). Știind că expresiile C/C++ de mai jos au ca valori id-ul, prima literă a numelui primului angajat, luna nașterii și anul angajării acestuia, scrieți definiția unei structuri cu eticheta **angajat**, care permite memorarea datelor despre angajații companiei, și declarați corespunzător variabila **p**.

p[0].Id **p[0].Nume[0]** **p[0].Data_N.luna** **p[0].Data_A.an** **(6p)**

3. 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 0. 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 de mai jos.

1	1	1	2	2	2
3	1	1	2	2	3
3	3	1	2	3	3
3	3	1	2	3	3
3	1	1	2	2	3
1	1	1	2	2	2

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

    ……….
```

SUBIECTUL al III-lea (30 puncte)

1. Scrieţi definiţia completă a unui subprogram **aranjare** care are doi parametri: **a** prin care primeşte un tablou unidimensional cu maximum 100 de numere întregi de maxim 4 cifre şi **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile de 2 cifre să fie ordonate descrescător, celelalte elemente din vector nefiind afectate de modificări. Tabloul modificat va fi furnizat tot prin intermediul parametrului a. Scrieţi definiţia completă a subprogramului aranjare. **(10p.)**

Exemplu: dacă tabloul are 6 elemente şi este de forma (12, -7, 61, -32, 800, 7),
după apel, acesta va fi: (61, -7, 12, -32, 800, 7).

2. Spunem că **un cuvânt t este derivat din cuvântul s** dacă s apare o singură dată în cuvântul t, s este prefix al lui t și t are cel puțin un caracter în plus față de s. De exemplu, cuvântul ***carte*** este derivat din cuvântul **car** dar cuvântul ***caricatura*** nu este derivat din cuvântul ***ca***. Un text are cel mult **100** de caractere și este format din cuvinte separate prin unul sau mai multe spații. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieţi un program C/C++ care citeşte de la tastatură textul și afișează pe ecran, separate între ele printr-un singur spațiu, cu majuscule cuvintele din text care derivă din primul cuvânt. Dacă textul nu conține nici un astfel de cuvânt se va afișa mesajul **NU EXISTA**.

Exemplu: pentru textul **el este acel elev care a fost eliminat iar ele sunt colegele lui**
se va afișa pe ecran: **ELEV ELIMINAT ELE**

(10p.)

3. Se consideră şirul 1, 1, 2, 1, 2, 3, 1,2, 3, 4… ... construit astfel: prima grupă este formată din numărul 1, a doua grupă este formată din numerele 1 şi 2, iar grupa a k-a, este formată din numerele 1, 2, ……k-1, k. Se cere să se citească din fișierul **bac.in** un număr natural n (n≤10000) și să se afișeze în fișierul **bac.out** cel de al n-lea termen al şirului dat. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă valoarea lui n este 10 se va afișa 4; dacă valoarea lui n este 12 se va afișa 2

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

b) Scrieţi programul C/C++ corespunzător algoritmului descris. **(8p.)**

REZOLVARE SUBIECTE TEST 9

SUBIECTUL I (20 de puncte)

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

SUBIECTUL al II – lea (40 de puncte)

1. a. 4

b. oricare 4 numere al căror cmmdc este 4. Un răspuns corect ar putea fi: 2020, 6060, 10100, 14140

c.

```
#include <iostream>
using namespace std;
int main()
{   int n,d,x,r;
    cin>>n;
    d=0;
    for(int i=1;i<=n;i++)
    {   cin>>x;
        if(d==0)
            d=x;
        else
        {
            do{
                r=x%d;
                x=d;
                d=r;
            } while(r);
            d=x;
        }
    }
    cout<<d;
    return 0;
}
```

d.

```
citește n (număr natural)
d←0
 i←1
dacă i<=n atunci
|   repetă
|       citește x (număr natural nenul)
|       dacă d=0 atunci
|       |       d←x
|       |  altfel
|       |       repetă
|       |       |       r←x % d
|       |       |       x←d
|       |       |       d←r
|       |       |_ până când r=0
|       |       d←x
|       |_■
|      i←i+1
|    până când i>n
|_■
scrie d
```

2. struct angajat{
unsigned int Id;
char Nume[51];
double salariu;
struct {
unsigned int zi, luna, an;
} Data_N, Data_A;
} p[100];

3. for(i=0;i<6;i++)
for(j=0;j<6;j++)
if((i<=j && i+j<=5) || (i>=j && i+j>=5))
//zona N sau zona S, inclusiv diagonalele
if (j<=2)
a[i][j]=1;
else
a[i][j]=2;
else
a[i][j]=3;//zonele E sau V

SUBIECTUL al III - lea (30 de puncte)

1.
```
void aranjare(int a[], int n)
{   for(int i=1;i<n;i++)
      if(a[i]/100==0&&a[i]/10!=0)
       for(int j=i+1;j<=n;j++)
          if(a[j]/100==0&&a[j]/10!=0)
             if(a[i]<a[j])
                swap(a[i],a[j]);
}
```

2.
```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{  int i,n,gasit=0;
   char s[101],*p, cuv[101];
  cin.getline(s,101);
  p=strtok(s," ");
  strcpy(cuv,p);  n=strlen(cuv);     //se reține în cuv primul cuvânt
  while(p)
  {
     if(strstr(p,cuv)==p && strstr(p+n,cuv)==0 && strlen(p)!=strlen(cuv)) //daca p derivă din cuv
     {
          for(int i=0;i<strlen(p);i++) //se transformă toate literele lui p în majuscule (se știe că toate
sunt litere mici)
           p[i]=p[i]-32;
        cout<<p<<' ';
        gasit=1;//se afișează cuvântul și se marchează că s-a găsit un cuvânt derivat din cuv
     }
     p=strtok(NULL," ");
  }
  if(!gasit)      cout<<"NU EXISTA";
   return 0;
}
```

3. Se împarte șirul 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, …… în grupe:
grupa 1: 1;
grupa 2: 1, 2;
grupa 3: 1, 2, 3;
…..
grupa k: 1, 2, ….k;

- Presupunem că termenul de rang n este ultimul din grupa completă k; în acest caz se obține relația 1+2+3+…..+k=n, adică k*(k+1)/2=n
- Numărul de grupe complete ale șirului până la termenul de rang n se obține ca soluție a ecuației k^2+k-2*n=0
- Se verifică dacă termenul de rang n este ultimul dintr-o grupă completă sau face parte dintr-o grupa incompletă
- Se determină poziția termenului de rang n în cadrul grupei lui și se afișează, aceasta fiind și valoarea termenului cerut

```
#include <iostream>
#include<cmath>
    using namespace std;
    int main()
    {   long long n,d,k,p;
       cin>>n;
    d=1+8*n;// se determină discriminantul ecuației de gradul II corespunzătoare șirului
     k=(-1+sqrt(d))/2; // se determină numărul de grupe complete  până la termenul de rang n
    if (n==k*(k+1)/2) //dacă termenul de rang n este ultimul din grupa completă k se afișează
                cout<<k;
      else
     {
      p=n-k*(k+1)/2; //se stabilește poziția termenului de rang n în cadrul grupei lui
    cout<<p;  //valoarea termenului este egală cu poziția lui în cadrul grupei lui
      }
    return 0;
      }
```

BAREM DE EVALUARE ȘI DE NOTARE TEST 9

- *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 puncte)

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

SUBIECTUL II (40 puncte)

1.	**a) Răspuns corect: 4**	6p	
	b) Răspuns corect: oricare 4 numere al căror cmmdc este 2020	6p	Un răspuns corect ar putea fi: 2020, 6060, 10100,14140
	c) Pentru program corect -declarare variabile -citire date -afişare date -instrucţiuni 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 instrucțiunile repetitive este corectă.
	d) Pentru algoritm pseudocod corect -echivalenţă a prelucrării realizate, conform cerinţei (*) -corectitudine globală a algoritmului[1])	**6p.** 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinţei, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	**Pentru rezolvare corectă** -definire a structurii/înregistrării (*) -declarare a variabilei conform cerinței -corectitudine globală a secvenței1)	**6p.** 3p. 2p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (definire principial corectă a unei structuri
3.	**Pentru rezolvare corectă** -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței	**6p.** 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (atribuire valori 1, atribuire valori 2, atribuire valori 3, elemente suport) conform cerinței.

SUBIECTUL al III - lea (30 puncte)

<table>
<tr><td>1.</td><td>Pentru subprogram corect
-antet subprogram (*)
-aranjare a elementelor în ordinea cerută (**)
-declarare a tuturor variabilelor locale, corectitudine globală a subprogramului</td><td>10p.
3p.
5p.

1p.
1p.</td><td>(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, cei 2 parametri) conform cerinței.
(**) Se acordă câte 2p. pentru fiecare aspect specific (identificare a elementelor cu exact 2 cifre, or onare descrescatoare, pastrarea celorlalte elemente nemodificate)</td></tr>
<tr><td>2.</td><td>Pentru program corect
-declarare a unei variabile care să memoreze un șir de caractere, declarare a variabilelor simple
-citire a datelor
-determinare a valorilor cerute (*)
-afișare a datelor (**)
- corectitudine globală a programului</td><td>10p.
1p.

1p.
4p.
3p.
1p.</td><td>(*) Se acordă câte 2p. pentru identificare corectă a entităților/cuvintelor, identificare corectă a cuvintelor derivate.
(**) Se acordă câte 1p. pentru afișare unui cuvântul derivat, 1p afișare cu majuscule, 1p pentru mesajul NU EXISTA</td></tr>
<tr><td rowspan="2">3.</td><td>a) Pentru răspuns corect
-coerenţă a descrierii algoritmului (*)
-justificare a elementelor de eficienţă</td><td>2p.
1p.
1p.</td><td rowspan="2">(*) 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 care nu folosește nicio structură repetitivă și care folosește doar variabile simple.
O soluţie posibilă citește valoarea lui n din fișier, stabilește numărul de grupe complete ale șirului, grupa din care face parte termenul de rang n, poziția termenului de rang n în cadrul grupei lui și valoarea termenului cerut
Un algoritm posibil de rezolvare este:
citeste n
d←1+8*n
k←(-1+sqrt(d))/2
daca n=k*(k+1)/2 atunci
| scrie k
| altfel
| p←n- k*(k+1)/2
| scrie p
|_■</td></tr>
<tr><td>b) Pentru program corect
-operaţii cu fișiere: declarare, pregătire în vederea citirii, citire din fișier
-determinare a valorii cerute (*),(**)
-utilizare a unui algoritm eficient (***)
-declarare a variabilelor, citire a datelor, corectitudine globală a programului</td><td>8p.
1p.

5p.
1p.

1p.</td></tr>
</table>