

MODEL TEST 9

- 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 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. Variabilele x , y și z sunt de tip întreg și memorează numere naturale din intervalul $[1, 10^3]$. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă x NU aparține reuniunii de intervale $[-5, -1] \cup [1, 5]$.

a. `!(x>=-5 && x<=-1) || !(x>=1 && x<=5)`

b. `!(x>=-5 || x<=-1 || x>=1 || x<=5)`

c. `x<-5 || x>5 || x>-1 && x<1`

d. `x<-5 && x>5 && x>-1 || x<1`

2. Pentru secvența de program alăturată, se dau mai jos patru triplete de numere. Fiecare dintre aceste triplete reprezintă un set de valori pentru variabila de intrare n . Care dintre triplete are proprietatea că, pentru toate cele trei valori ale lui n din triplet se obține aceeași valoare a lui S ?

```
S=0; i=3;
while (i<=n)
{ S=S+i;
  i=i+3;
}
cout<<S; | printf("%d",S);
```

a. (3, 5, 6)

b. (6, 7, 8)

c. (10, 11, 12)

d. (6, 9, 12)

2. Variabilele x și y sunt de tip real. O transcriere în limbajul C/C++ a expresiei alăturate este:

$$\sqrt{x^2 + 1}$$

a. `sqrt((1+x)*x)`

b. `1+sqrt(x)`

c. `sqrt(1+x*x)`

d. `(1+x*x)1/2`

3. Deduceți care vor fi elementele vectorului a la sfârșitul execuției secvenței de program alăturate, pentru cazul în care $n=7$ și elementele vectorului v sunt, în ordine (0, 2, 7, 3, 4, 8, 5).

```
j=0;
for(i=0;i<n;i++)
if(v[i]%2==0 && i%2!=0)
{ a[j]=v[i]; j=j+1;}
```

a. (3)

b. (2, 8)

c. (0, 4)

d. (0,7, 4, 5)

4. Vectorul v are 7 elemente întregi $(-1, 2, -3, 4, 5, -6, 7)$, i și a sunt variabile întregi. După executarea secvenței de program alăturate vectorul v este:

```
a=0;
for (i=0; i<=6; i++)
    if (a>v[i]){a=v[i]; v[i]=0;}
    else v[i]++;
```

a. 0 3 -2 0 0 -5 0

b. 0 3 0 5 6 0 8

c. 0 0 -2 0 0 -5

d. 0 3 -2 0 0 -5 8

SUBIECTUL al II-lea**(40 puncte)**

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

- a. Ce se va afișa dacă se citește 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 dat. (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
        pînă când r=0
    d←x
scrie d
```

2. Variabila a memorează un număr natural. Scrieți o secvență de instrucțiuni C++ în urma executării căreia să se afișeze pe ecran suma cifrelor impare din numărul respectiv. (6p.)

3. Fără a utiliza alte variabile, înlocuiți punctele de suspensie din secvența de mai jos astfel încât, în urma executării ei, să se insereze pe poziția p numărul întreg, v , într-un șir s cu n numere întregi, cu elemente numerotate de la 1. (de exemplu, dacă șirul s are elementele 2,4,6,8,10, valoarea $v=12$ și dacă $p=3$ șirul s va avea după execuția secvenței elementele: 2, 4, 12, 6, 8, 10.

```
cin>>p;
cin>>v;
n=n+1;
.....
s[i]=s[i-1]
.....
```

(6p.)

SUBIECTUL al III-lea**(30 de puncte)**

1. Scrieți un program care citește de la tastatură un număr natural n , care reprezintă numărul de elemente ale unui tablou unidimensional a cu maximum 100 de numere întregi și apoi cele n elemente ale tabloului. Programul 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 afișat pe ecran, elementele fiind separate printr-un singur spațiu.

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

(10p.)

2. Scrieți în pseudocod algoritmul de rezolvare al problemei următoare: se citește de la tastatură un număr natural n și afișează suma divizorilor primi din descompunerea numărului în factori primi.

Exemplu: dacă $n=1540$ se va afișa $S=2+5+7+11=25$. ($1540=2^2 \cdot 5 \cdot 7 \cdot 11$)

(10p.)

3. Se consideră șirul 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 și 2,.... 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 ($n \leq 1000$) și să se afișeze în fișierul **bac.out** cel de al n -lea termen al șirului 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) Scrieți programul C/C++ corespunzător algoritmului descris.

(8p.)

REZOLVARE SUBIECTE
TEST 9
SUBIECTUL I

(20 de puncte)

| | | | | | |
|-----|-----|-----|-----|-----|------|
| 1 c | 2 b | 3 c | 4 b | 5 b | 5x4p |
|-----|-----|-----|-----|-----|------|

SUBIECTUL al II – lea

(40 de puncte)

1.a. 4

b. oricare 4 numere al căror cmmdc este 2020. 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.

```

b=0;
cin >> a;
while (a != 0)
{
  if(a%10%2==1)
    b = b + a % 10;
  a = a / 10;
}
cout << b;

```

3.

```

cin>>p;
cin>>v;
n=n+1;
for (i=n;i>=p+1; i--)
    s[i]=s[i-1];
s[p]=v;

```

SUBIECTUL al III - lea**(30 de puncte)**

```

1. #include <iostream>
    using namespace std;
int main()
{
int a[101], int n, aux;
cin>>n;
for(int i=1;i<=n;i++)
    cin>>a[i];
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])
                    { aux=a[i];
                      a[i]=a[j];
                      a[j]=aux;}
for(int i=1;i<=n;i++)
    cout<<a[i]<<" ";
return 0;}

```

2.

```

citeste n;
d←2; S←0;
cat timp (n>1) executa
| nr←0;
| cat timp (n%d=0) executa
| | n←n/d;
| | nr←nr+1;
| | ■
| | daca (nr>0) atunci
| | | S=S+d;
| | ■
| | d←d+1;
| | ■
scrie S;

```

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 $1+2+3+\dots+k=n$, 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 din șir 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 din care face parte
```

```
}
```

```
return 0;
```

```
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 9**

- 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)**

| | | |
|----------------------------|-------------|--|
| 1 c 2 b 3 c 4 b 5 b | 5x4p | |
|----------------------------|-------------|--|

SUBIECTUL al II – lea**(40 de puncte)**

| | | | |
|----|--|--|--|
| 1. | a) Răspuns corect: 4 | 6p | |
| | b) Răspuns corect: oricare 4 numere al căror cmmdc este 2020 | 6p | |
| | c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie -instrucțiuni 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 -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾ | 6p. 5p. 1p. | (*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței. |
| 2. | Pentru rezolvare corectă - algoritm corect de spargere a numărului în cifre - identificare cifră impară -determinare a valorii cerute - afișare a rezultatului -corectitudine sevență | 6p. 2p. 1p. 1 p 1p 1p | |
| 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)**

| | | | |
|----|---|--|---|
| 1. | Pentru program corect -declarare variabile -citire vector -aranjare a elementelor în ordinea cerută (*) -afisare elemente -corectitudine program | 10p. 1p. 1p. 6p. 1p. 1p. | (*)Se acordă câte 2p. pentru fiecare aspect specific (identificare a elementelor cu exact 2 cifre, ordonare descrescătoare, pastrarea celorlalte elemente nemodificate) |
| 2. | Pentru program corect -citire număr -determinare a valorii cerute (*) -afișare a rezultatului - corectitudine globală algoritm | 10p. 2p. 6p 1p. . 1p. | (*) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unui divizor, identificare a unui divizor prim, calculare a sumei) |
| 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, citire a datelor, corectitudine globală a programului1) | 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 care nu folosește nicio structură repetitivă și care folosește doar variabile simple. O soluție 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 Un algoritm posibil de rezolvare este: Citeste n $d \leftarrow 1 + 8 * n$ $k \leftarrow (-1 + \text{sqrt}(d)) / 2$ daca $n = k * (k + 1) / 2$ atunci scrie k altfel $p \leftarrow n - k * (k + 1) / 2$ scrie p ■ |