

MODEL TEST 10

- 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 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. În secvența de program de mai jos, instrucțiunea de afișare se va executa de un număr de ori egal cu:

```
for (i=1;i<=4;i++);
  for (j=i+1;j<=10;j++)
    cout<<j; | printf("%d",j);
```

a. 24

b. 6

c. 5

d. 30

(4 p)

2. Subprogramul f este definit alăturat. Indicați ce se afișează în urma apelului f(12345).

```
void f (int x)
{ cout<<" "; | printf("+");
  if(x>0) { f(x/100);
  cout<<x; | printf("%d",x); }
  cout<<" "; | printf("+");
}
```

a. +++++1+123+12345+

b. +12345+123+1+++++

c. ++++112312345

d. +1+123+12345+

(4 p)

3. Utilizând metoda backtracking, sunt generate în ordine lexicografică toate anagramele cuvântului **house** astfel încât două vocale să nu fie alăturate. Să se precizeze câte anagrama se generează între soluțiile **esuho** și **osehu**.

a) 3

b) 0

c) 2

d) 1

(4 p)

4. Precizați care dintre următoarele grafuri orientate cu patru vârfuri date prin vectorul de arce este tare conex.

a) $U = ((1, 2), (1, 4), (3, 2), (3, 1))$ b) $U = ((1, 2), (1, 4), (2, 3), (3, 1), (4, 3))$ c) $U = ((1, 2), (1, 4), (1, 3))$ d) $U = ((1, 2), (2, 4), (4, 1))$

(4p)

5. Se consideră graful neorientat cu 7 noduri, reprezentat prin următoarea listă de muchii: [1, 7], [1, 2], [2, 7], [7, 6], [2, 6], [2, 5], [3, 4], [2, 3], [5, 6].

Să se precizeze numărul minim de muchii ce trebuie adăugate astfel încât graful să devină eulerian.

a) 3

b) 2

c) 4

d) 0

(4 p)

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

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

- a) Scrieți ce valoare va fi afișată în urma execuției algoritmului dacă se citesc, în această ordine, valorile 11, 57, 9. (6 p)
- b) Dacă pentru z se citește valoarea 17, dați un exemplu de valori pentru a și b astfel încât să se afișeze două valori nule 0 0. (6 p)
- c) Să se scrie programul C/C++ corespunzător algoritmului dat. (10 p)
- d) Să se scrie un algoritm echivalent cu cel dat în care să nu se folosească structuri repetitive. (6p)

Citește a, b, c (numere naturale nenule)┌dacă $a > b$ atunci| $a \leftarrow a + b$ | $b \leftarrow a - b$ | $a \leftarrow a - b$

└─┘

 $s \leftarrow 0$ $n \leftarrow 0$ ┌pentru $x \leftarrow b, a, -1$ execută| ┌dacă $x \% c = 0$ atunci| | $n \leftarrow n + 1$ | | $s \leftarrow s + x$

| └─┘

└─┘

Scrie $n, ', s$

2. Se consideră următoarea declarație:

```
struct cerc
{ float r;
  struct
  {int x,y;}centru;
  }c1,c2;
```

Scrieți o expresie C++ care să fie adevărată dacă și numai dacă cele două cercuri $c1$ și $c2$, de centre diferite, nu se intersectează. (6p.)

3. În secvența următoare, a este un tablou bidimensional cu 5 linii și 5 coloane. Fără a utiliza alte variabile, completați secvența cu instrucțiunile care generează elementele tabloului următor:

```
for(i=0;i<5;i++)
for(j=0;j<5;j++)
.....
```

```
25 24 23 22 21
20 19 18 17 16
15 14 13 12 11
10 9 8 7 6
5 4 3 2 1
```

(6p.)

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

1. Subprogramul **comune** are doi parametri, a și b , prin care primește câte un număr natural ($a \in [0, 10^9]$, $b \in [0, 10^9]$). Subprogramul returnează cel mai mare număr care se poate forma cu cifrele distincte comune celor două valori, sau valoarea -1 în cazul în care cele două valori nu au nicio cifră comună. Scrieți definiția completă a subprogramului.

Exemplu: Dacă $a=806528$ și $b=207068$, subprogramul returnează numărul 8620 .

(10p.)

2. Se citește de la tastatură un șir de caractere, format din maximum 250 de litere mari și mici ale alfabetului englez. Cuvintele textului sunt separate între ele prin caracterele spațiu, '.' și ', '. Să se afișeze pe ecran toate cuvintele din textul citit care conțin același număr de vocale ca și ultimul cuvânt, sau mesajul **“nu există”**, dacă niciun cuvânt din text nu îndeplinește această condiție. Cuvintele vor fi scrise fiecare pe câte o linie de ecran.

Exemplu: Dacă se citește șirul **Scoala online va deveni un mod de instruire obisnuit pentru elevi.**, se vor afișa pe ecran:

Scoala
online
deveni

(10p.)

3. Fișierul **“bac.txt”** conține un șir de cel mult 10^9 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran în ordine **crescătoare**, separate cu spațiu, toate valorile de două cifre aflate în șirul citit din fișier, care apar de cele mai multe ori ca prefixe ale numerelor cu mai mult de două cifre aflate în fișier. Dacă nu există astfel de valori, se va afișa pe ecran mesajul **“nici o valoare”**.

Proiectați un algoritm eficient din punctul de vedere al timpului de execuție.

Exemplu: Dacă fișierul bac.txt conține valorile 1234 **78** **12** 978 132 128 7800 222 97 7831 13, se afișează pe ecran valorile 12 78.

a. Scrieți programul C/C++ corespunzător algoritmului proiectat.

(8p.)

b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia.

(2p.)

REZOLVARE SUBIECTE
TEST 10
SUBIECTUL I
(20 de puncte)

1. c) 2. a) 3. c) 4. b) 5. b)

SUBIECTUL al II-lea
(40 de puncte)

1.

a) 5 180

b) Orice doua valori ce definesc extremitatile unui interval ce nu contine niciun multiplu de 17

c)

```
#include <iostream>
using namespace std;
int a,b,c,x,n,s;
int main()
{
    cin>>a>>b>>c;
    if(a>b)
    {
        a=a+b;
        b=a-b;
        a=a-b;
    }
    for(x=b;x>=a;x--)
        if(x%c==0)
        {
            n++;
            s=s+x;
        }
    cout<<n<<' '<<s;
    return 0;
}
```

d)

citește a, b, c (numere naturale nenule)

┌dacă $a > b$ atunci

| $a \leftarrow a + b$

| $b \leftarrow a - b$

| $a \leftarrow a - b$

└■

$n \leftarrow b/c - (a-1)/c$

┌dacă $a \% c \neq 0$ atunci

| $a \leftarrow (a/c + 1) * c$

└■

$b \leftarrow b/c * c$

$s \leftarrow (a + b) * n / 2$

Scrie n, ' ', s

2. $\sqrt{\text{pow}(c1.\text{centru}.x - c2.\text{centru}.x, 2) + \text{pow}(c1.\text{centru}.y - c2.\text{centru}.y, 2)} > c1.r + c2.r$

3. for(i=1; i<=5; i++)

for(j=1; j<=5; j++)

a[i][j]=5*(4-i) +(5- j);

SUBIECTUL al III-lea

(30 de puncte)

1.

```
int comune(int a, int b)
{
    int va[10]={0},vb[10]={0},x=0,c,k=0;
    while(a)
    {
        va[a%10]++;
        a=a/10;
    }
    while(b)
    {
        vb[b%10]++;
        b=b/10;
    }
    for(c=9;c>=0;c--)
        if(va[c]&&vb[c])
        {
            x=x*10+c;
            k++;
        }
    if(k)
        return x;
    return -1;
}
```

2.

```
#include <bits/stdc++.h>
using namespace std;
char s[251],*p,v[200][25],voc[]="aeiouAEIOU";
int n, i, nvi,nvu,ok,j;
```

```
int main()
{
    cin.getline(s,251);
    p= strtok(s, " ,.");
    while(p)
    {
        strcpy(v[++n],p);
        p= strtok(NULL, " ,.");
    }
    for(i=0;i<strlen(v[n]);i++)
        if(strchr(voc,v[n][i]))
            nvu++;
    for(i=1;i<n;i++)
    {
        nvi=0;
        for(j=0;j<strlen(v[i]);j++)
            if(strchr(voc,v[i][j]))
                nvi++;
        if(nvi==nvu)
        {
            cout<<v[i]<<endl;
            ok=1;
        }
    }
}
```

```

}
if(!ok)
  cout<<"nu exista" ;
return 0;
}

```

3.a)

```

#include <bits/stdc++.h>
using namespace std;
ifstream fin("bac.txt");

int a[100],b[100],x,maxi, ok;

int main()
{
  while(fin>>x)
  {
    if(x>9&&x<100)
      a[x]++;
    else
      if(x>99)
      {
        while(x>99)
          x=x/10;
        b[x]++;
      }
  }
  for(x=10;x<100;x++)
    if(a[x]&&b[x])
      if(a[x]>maxi)
        maxi=a[x];
  for(x=10;x<100;x++)
    if(a[x]&&b[x])
      if(a[x]==maxi)
      {
        cout<<x<<' ';
        ok=1;
      }
  if(!ok)
    cout<<"nici o valoare";
  return 0;
}

```

b) Algoritmul citește secvențial valorile, contorizează frecvența numerelor de două cifre în vectorul *a*, iar prefixele de două cifre sunt contorizate în vectorul *b*; După determinarea maximului de apariții, se parcurg pozițional și se afișează valorile care îndeplinesc cerința enunțului. Algoritmul este eficient ca timp de execuție deoarece face determinările cerute odată cu citirea datelor, deci este liniar.

BAREM DE EVALUARE ȘI DE NOTARE**TEST 10**

- 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 al III-lea**(20 de puncte)**

1.c) ; 2.a) ; 3. c) ; 4. b) ; 5. b)	5x4p
--	------

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

1	a) Răspuns corect: 5 180	6p.	Se acordă câte 3p. pentru fiecare dintre cele două valori conform cerinței.
	b) Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței (orice două valori care nu conțin, în intervalul delimitat, niciun multiplu de 17).
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie (*) -instrucțiune repetitivă -atribuiri -corectitudine globală a programului ¹⁾	10p 1 p. 1p. 1p 3p 2p 1p 1p	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile de decizie este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței -corectitudine globală a algoritmului ¹⁾	6p.	Se acordă 2p pentru determinarea numărului de multipli de c din intervalul [a,b] și 4p pentru calculul sumei progresiei aritmetice de rație c, termen inițial a (adus la valoarea primului multiplu de c din interval) și termen final b (adus la valoarea ultimului multiplu de c din interval)
2	Pentru rezolvare corectă -referire corectă la campurile structurii/înregistrării -utilizarea corectă a formulei pentru distanța dintre centre -corectitudine globală a secvenței ¹⁾	6p. 2p 3p 1p	Condiția cerută va verifica dacă distanța dintre centrele cercurilor este mai mare decât suma razelor
3	Pentru rezolvare corectă -acces la un element al tabloului -atribuire corectă a valorilor, conform cerinței (*) -corectitudine globală a secvenței ¹⁾	6p. 1p 4p 1p	(*) Se poate folosi , de exemplu, expresia de atribuire $5*(4-i) + (5-j)$, sau orice altă regulă corectă de atribuire.

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

1.	Pentru subprogram corect -antet subprogram (*) -determinare a valorii cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului -declaraire a tuturor variabilelor locale, corectitudine globală a subprogramului ¹⁾	10p . 2p. 6p. 1p. 1p	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (determinare cifre comune, formarea valorii rezultat, tratarea cazului particular).
2.	Pentru program corect -declaraire a unei variabile care să memoreze un sir de caractere si a unui vector de siruri -citirea datelor -separarea cuvintelor din text, cu memorarea lor(*) - identificarea cuvintelor care respectă cerinta (**) -afisarea rezultatelor(tratarea cazului “nu există”) -declaraire a variabilelor simple, corectitudine globală a programului ¹⁾	10p . 1p. 1p. 2p. 4p. 1p 1p	(*) Se acordă câte 1p. pentru fiecare aspect specific (identificare unui cuvânt, memorarea prin copiere) (**) Determinarea numărului de vocale din ultimul cuvânt, determinarea numărului de vocale din fiecare cuvânt analizat.
3.	a) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență b) Pentru program corect -operații cu fișiere: declaraire, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizare a unui algoritm eficient (***) -declaraire a variabilelor, citire 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 . O soluție posibilă citește valorile din fișier, numără valorile de două cifre într-un vector de frecvență , numără prefixele de două cifre ale numerelor ≥ 100 în alt vector de frecvență, parcurge apoi pozitional cei doi vectori, determinând numărul maxim de aparitii(dintre acele valori x care au fost numarate în ambii vectori) și marcând existenta unor astfel de valori. Afisarea se face prin parcurgerea si identificarea valorilor care au numărul de aparitii egal cu valoarea maximă determinată.