

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. Expresia C/C++

$(x \geq 18) \ \&\& \ ! (x < 19 \ || \ x > 20) \ \&\& \ (x \leq 21)$

are valoarea 1 dacă și numai dacă valoarea memorată de variabila întreagă x aparține intervalului:

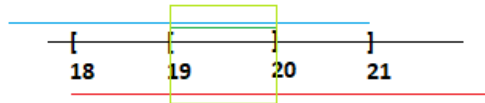
a. [18,20]

b. [18,21]

c. [19,20]

d. [19,21]

$(x \geq 18) \ \&\& \ ! (x < 19 \ || \ x > 20) \ \&\& \ (x \leq 21)$



2. Utilizând metoda backtracking se generează toate posibilitățile de a planta de-a lungul unei alei cinci arbuști decorativi din mulțimea {caprifoi, iasomie, liliac, tamarix, scumpie}. Două soluții sunt diferite dacă ordinea arbuștilor diferă. Primele patru soluții obținute sunt, în această ordine: (caprifoi, iasomie, liliac, tamarix, scumpie), (caprifoi, iasomie, liliac, scumpie, tamarix), (caprifoi, iasomie, tamarix, liliac, scumpie), (caprifoi, iasomie, tamarix, scumpie, liliac). Indicați penultima soluție generată.

a. (scumpie, tamarix, caprifoi, iasomie, liliac)

b. (scumpie, tamarix, caprifoi, liliac, iasomie)

c. (scumpie, tamarix, liliac, caprifoi, iasomie)

d. (scumpie, tamarix, liliac, iasomie, caprifoi)

3. Fiecare dintre variabilele A și B , declarate alăturat, memorează coordonatele (x abscisa, iar y ordonata) câte unui punct în sistemul de coordonate xOy . Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă segmentul cu extremitățile în punctele corespunzătoare variabilelor A și B intersectează axa Oy a sistemului de coordonate.

```
struct punct  
{ int x,y;  
  } A,B;
```

a. $A.x * B.x \leq 0$

b. $A(x) * B(x) \leq 0$

c. $x.A * x.B \leq 0$

d. $punct(A,B).y == 0$

4. Într-un arbore cu rădăcină un nod se află pe nivelul x dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea x . Pe nivelul 0 se află un singur nod (rădăcina).

Un arbore cu rădăcină are 8 noduri, numerotate de la 1 la 8, și muchiile [1,3], [1,7], [1,8], [2,4], [3,5], [3,6], [4,5]. Știind că rădăcina arborelui este nodul numerotat cu 5, indicați nodurile situate pe nivelul 2 al arborelui dat.

a. 7,8

b. 7,4

c. 1,4,6

d. 1,2,6

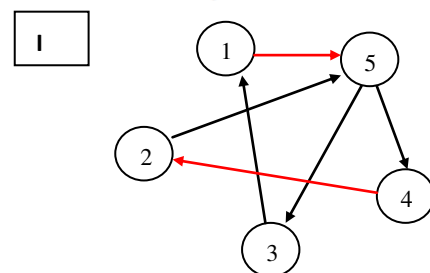
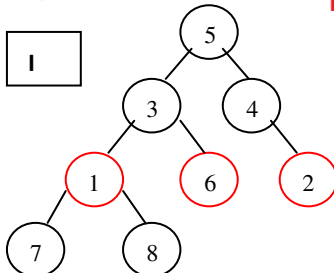
5. Un graf orientat cu 5 vârfuri, numerotate de la 1 la 5, are arcele (2,5), (3,1), (5,3), (5,4). Indicați numărul minim de arce care trebuie adăugate acestuia, astfel încât graful obținut să fie tare conex.

a. 1

b. 2

c. 3

d. 4



SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

- a. Scrieți numărul afișat în urma executării algoritmului dacă pentru n se citește valoarea 7. **4** (6p.)
- b. Scrieți două numere din intervalul $[10, 10^2)$ care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 10. **19, 20** (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat una dintre structurile **cât timp...execută** cu o structură repetitivă de tip **pentru...execută**. (6p.)

```

citește n (număr natural nenul)
nr ← 0; i ← 1
cât timp i ≤ n execută
    x ← 0; y ← 1; j ← 1
    cât timp j < i execută
        r ← 2*x-y; x ← y; y ← r
        j ← j+1
    i ← i+1
    dacă y > 0 atunci
        nr ← nr+1
scrie nr

```

```

I
#include <iostream>
using namespace std;
unsigned int n; int
nr,i,x,y,j,r;
int main()
{
    cin >> n;
    nr=0; i=1;
    while(i <= n)
    {
        x=0; y=1; j=1;
        while(j < i)
        {
            r=2*x-y; x=y; y=r;
            j=j+1;
        }
        i=i+1;
        if(y > 0)
            nr=nr+1;
    }
    cout << nr;
    return 0;
}

```

```

II
citește n (număr natural nenul)
nr ← 0; i ← 1
cât timp i ≤ n execută
    x ← 0; y ← 1; j ← 1
    cât timp j < i execută
        r ← 2*x-y; x ← y; y ← r
        j ← j+1
    i ← i+1
    dacă y > 0 atunci
        nr ← nr+1
scrie nr

```

```

citeste n
nr=0; i=1
cat timp (i <= n) executa
    x=0; y=1
    pentru j=1, i-1 executa
        r=2*x-y; x=y; y=r;
    i=i+1
    daca (y > 0) atunci
        nr=nr+1
scrie nr

```

2. Subprogramul f este definit alăturat. Scrieți două numere naturale din intervalul deschis $[1, 10]$, care pot fi memorate în variabilele întregi x_1 , respectiv x_2 , astfel încât valoarea lui $f(10, x_1)$ să fie 5, iar valoarea lui $f(x_2, 10)$ să fie 1. **$x_1=6, x_2=10$** (6p.)

```

int f(int a, int b)
{
    if(a > b) return a/b+f(a-b,b);
    if(a < b) return b/a+f(a,b-a);
    return 1;
}

```

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 4 linii și 5 coloane, numerotate începând de la 0, cu elemente numere întregi, inițial toate nule. Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni astfel încât, în urma executării acestora, variabila a să memoreze tabloul alăturat. (6p.)

```

16 17 18 19 20
11 12 13 14 15
6 7 8 9 10
1 2 3 4 5

```

```

for(i=3; i>=0; i--)
    for(j=0; j<5; j++)
        if(i==3) a[i][j]=j+1;
        else a[i][j]=a[i+1][j]+5;

```

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

1. Subprogramul **prodprim** are doi parametri:

- **n**, prin care primește un număr natural ($n \in [2, 10^9]$);
- **p**, prin care furnizează produsul divizorilor primi ai lui **n**.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=2000**, în urma apelului **p=10**, deoarece $2000=2^4 \cdot 5^3$.

(10p.)

```
int prim (unsigned long n)
{
    unsigned long i;
    for(i=2;i<=n/2;i++)
        if(n%i==0) return 0;
    return 1;
}

void prodprim(unsigned long n, unsigned long &p)
{
    unsigned long i;
    p=1;
    for(i=2;i<=n;i++)
        if(prim(i)==1 && n%i==0) p=p*i;
}
```

2. Într-un text cu cel mult 100 de caractere, cuvintele sunt formate 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 menționat și afișează pe ecran, pe linii separate, toate cuvintele sale pentru care numărul de vocale este strict mai mic decât numărul de consoane. Dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele din mulțimea **a, e, i, o, u**.

Exemplu: pentru textul **ei au plantat tamarix ea a adus iasomie** se afișează pe ecran, nu neapărat în această ordine, cuvintele alăturate.

(10p.)**plantat
tamarix**

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

int main()
{
    char str[101], sir[101];
    char *p;
    int i, n; char v[50][20];
    cin.get(str, 100);
    n=0;
    p = strtok(str, " ");
    while( p != NULL )
    {
        n++; strcpy(v[n], p);
        p = strtok(NULL, " ");
    }

    int c, w, j, k=0;
    for(i=1; i<=n; i++)
    {
        c=0; w=0;
        for(j=0; j<strlen(v[i]); j++)
            if(strchr("aeiou", v[i][j])!=0) w++;
            else c++;
        if(w<c) {cout<<v[i]<<endl; k++;}
    }
    if(k==0) cout<<"nu exista";
    return 0;
}
```

3. Se citesc de la tastatură două numere naturale din intervalul **[1, 81]**, **p1** și **p2**, și se cere scrierea în fișierul **bac.out** a tuturor numerelor naturale cu exact 7 cifre, pentru care produsul primelor două cifre este egal cu **p1**, cele trei cifre din mijloc sunt egale între ele, iar produsul ultimelor două cifre este egal cu **p2**. Numerele apar în fișier în ordine strict descrescătoare, fiecare pe câte o linie. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă **p1=12**, iar **p2=8**, atunci **2633324** și **3400018** sunt două dintre cele 160 de numere cu proprietatea cerută ($2 \cdot 6 = 3 \cdot 4 = 12$ și $2 \cdot 4 = 1 \cdot 8 = 8$).

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

(8p.)

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

(2p.)

```
#include <iostream>
#include <fstream>
using namespace std;

ofstream f("bac.out");
```

```

int main()
{
unsigned long c1, c2, c3,i,j,k,p1,p2,s,s1;
cin>>p1>>p2;
for(i=9;i>=1;i--)
{
    if(p1%i==0 && p1/i<=9)
    {
        c1=i*10+(p1/i);
        for(j=0;j<=9;j++)
        {
            c2=j*100+j*10+j;
            s1=c1*1000+c2;
            for(k=9;k>=1;k--)
            {
                if(p2%k==0 && p2/k<=9) {
                    c3=k*10+(p2/k);
                    s=s1*100+c3;
                    f<<s<<endl;
                    // cout<<s<<' ';
                }
            }
        }
    }
}
return 0;
}

```