

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.

- Indicați o expresie care are valoarea 1 dacă și numai dacă numărul memorat în variabila întreagă x aparține intervalului închis $[-20, 20]$.
 - $!(x < -20) \ || \ (x > 20)$
 - $x > -20 \ || \ x < 20$
 - $!(x + 20 < 0 \ \&\& \ x - 20 > 0)$
 - $x + 20 \geq 0 \ \&\& \ x - 20 \leq 0$
- Subprogramul f este definit alăturat. Valoarea lui $f(2020, 15)$ este:


```

int f(int x, int y)
{
    if(x==0) return y;
    if(y==0) return x;
    return f(x-y, x*y);
}
      
```

 - 14
 - 200
 - 1990
 - 2020
- Utilizând metoda backtracking se generează toate grupele de accesorii pentru tenis de câmp din mulțimea {bentiță, fileu, grip, manșete, mingi, rachetă, racordaj, șapcă}. Accesoriiile au prețurile următoare, exprimate în lei: bentiță - 40, fileu - 400, grip - 30, manșete - 30, mingi - 10, rachetă - 400, racordaj - 70, șapcă - 60. Într-o grupă accesoriiile sunt distincte, nu contează ordinea lor și costă, în total, exact 500 de lei. Primele trei soluții generate sunt, în această ordine: (bentiță, fileu, grip, manșete), (bentiță, fileu, șapcă), (bentiță, grip, manșete, rachetă). A cincea soluție generată este:
 - (bentiță, rachetă, șapcă)
 - (fileu, grip, mingi, șapcă)
 - (grip, rachetă, racordaj)
 - (manșete, mingi, rachetă, șapcă)
- 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). Într-un arbore cu rădăcină cu 6 de noduri, numerotate de la 1 la 6, sunt 3 noduri situate pe ultimul nivel. Indicați un vector de „tați” care poate corespunde arborelui.
 - (0, 3, 1, 6, 6, 2)
 - (2, 0, 1, 1, 2, 1)
 - (3, 5, 5, 6, 0, 5)
 - (5, 1, 1, 1, 0, 1)
- Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, are arcele (1, 6), (2, 1), (2, 3), (2, 5), (3, 4), (3, 5), (4, 6). Indicați numărul minim de arce care pot fi adăugate astfel încât graful obținut să aibă cel puțin două circuite. Un circuit este format numai din arce distincte, iar două circuite sunt distincte dacă diferă prin cel puțin un arc.
 - 1
 - 2
 - 3
 - 4

SUBIECTUL al II-lea

(40 de puncte)

- 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 .
 - Scrieți valoarea afișată dacă se citește numărul 25940464. (6p.)
20950
 - Scrieți trei numere de cinci cifre care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului să se afișeze 2020. (6p.)
Numere de forma $x0y0z$,
unde x, y și z sunt cifre pare diferite de 0
 - Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
 - Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura **repetă...până când** cu o structură repetitivă cu test inițial. (6p.)

```

citește n (număr natural)
m ← 0
repetă
    c ← n % 10; n ← [n/10]
    dacă c = 0 atunci c ← 2
    altfel
        dacă c % 2 = 0 atunci
            c ← 0
        ■
        m ← m * 10 + c
până când n = 0
scrie m
      
```

d) <pre> citește n (număr natural) m ← 0 repetă cat timp n! = 0 execută c ← n % 10; n ← [n/10] dacă c = 0 atunci c ← 2 altfel dacă c % 2 = 0 atunci c ← 0 ■ m ← m * 10 + c ■ până când n = 0 scrie m </pre>	c) <pre> #include <iostream> using namespace std; int main() { unsigned long n,m,c; cin >> n; m = 0; do { c = n % 10; n = n / 10; if (c == 0) c = 2; else if (c % 2 == 0) c = 0; m = m * 10 + c; } while (n != 0); cout << m; return 0; } </pre>
--	--

2. Variabila **s** memorează simultan următoarele date despre fiecare dintre cele 20 de specii de animale dintr-o rezervație: un cod, reprezentând specia, numărul de exemplare din specia respectivă și vârstele acestora. În rezervație sunt maximum 10 exemplare din fiecare specie. Știind că expresiile C/C++ de mai jos au ca valori numere naturale și reprezintă codul și numărul de exemplare din prima specie, respectiv vârsta celui de al 4-lea exemplar din această specie, scrieți definiția unei structuri cu eticheta **specie**, care permite memorarea datelor despre o specie, și declarați corespunzător variabila **s**.

s[0].cod s[0].nrExemplare s[0].varsta[3] (6p.)

```

struct specie {
    int cod, nrExemplare;
    int varsta[11];
} s[21];

```

3. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu 5 linii și 7 coloane, numerotate începând de la 0, cu elemente numere întregi, inițial toate egale cu 1.

Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni prin care se transformă în memorie tabloul, astfel încât orice element aflat pe prima linie sau pe prima coloană să aibă valoarea 1 și oricare alt element din tablou să fie egal cu ultima cifră a sumei celor două elemente alăturate lui, aflate pe aceeași linie dar pe coloana din stânga, respectiv pe aceeași coloană, dar pe linia anterioară. **(6p.)**

1	1	1	1	1	1	1
1	2	3	4	5	6	7
1	3	6	0	5	1	8
1	4	0	0	5	6	4
1	5	5	5	0	6	0

```

for(i=0;i<=4;i++)
    for(j=0;j<=6;j++)
        a[i][j]=1;
for(i=1;i<=4;i++)
    for(j=1;j<=6;j++)
        a[i][j]=(a[i-1][j]+a[i][j-1])%10;

```

SUBIECTUL al III-lea

(30 de puncte)

1. Două numere distincte **a** și **b** sunt numite **d-fii** ai unui număr natural **n** dacă **a · b = n**. Subprogramul **fii** are un singur parametru, **n**, prin care primește un număr natural ($n \in [2, 10^9]$). Subprogramul afișează pe ecran toate perechile distincte de numere naturale cu proprietatea că sunt d-fii ai lui **n**. Fiecare pereche este afișată încadrată între paranteze rotunde, numerele din pereche fiind afișate în ordine strict descrescătoare, separate printr-un spațiu. Scrieți definiția completă a subprogramului.
- Exemplu:** dacă **n=12** se afișează pe ecran, neapărat în această ordine, (12 1) (6 2) (4 3) iar dacă **n=16** se afișează pe ecran (16 1) (8 2) **(10p.)**

```

void fii(unsigned long n)
{
    cout<<'('<<n<<' '<<1<<');
    int min=2;
    for(int i=n/2;i>=min;i--)
        if(n%i==0 && n/i!=i) {
            cout<<'('<<i<<' '<<n/i<<');min=n/i+1;}
}

```

2. Un text are cel mult 100 de caractere și este format din cuvinte și numere, separate prin câte un spațiu. Cuvintele sunt formate numai din litere ale alfabetului englez. Toate numerele sunt reale și sunt formate numai din parte întreagă sau din parte întreagă și parte fracționară, separate prin virgulă (,), numerele negative fiind precedate de semnul minus (-).

Scrieți un program C/C++ care citește de la tastatură textul, pe care îl transformă, eliminând din componența sa toate numerele negative. Programul afișează apoi pe ecran textul obținut.

Exemplu: pentru textul

2,7 minus 3,5 minus 2 egal 2,7 plus -3,5 plus -2 egal -0,2 rezultat

se va afișa pe ecran textul:

2,7 minus 3,5 minus 2 egal 2,7 plus plus egal rezultat

(10p.)

```

#include <iostream>
#include <string.h>
#include <stdlib.h>
using namespace std;
int main()
{
    char str[101],sir[101];
    char *p;
    int i,n; char v[50][20],k;
    cin.get(str,100);
    n=0;
    p = strtok(str, " ");
    while( p != NULL )
    {
        n++; strcpy(v[n],p);
        p = strtok(NULL, " ");
    }
    strcpy(sir,"");
    for(i=1;i<=n;i++)
    {
        if(v[i][0]!='-')
        {
            strcat(sir, " ");
            strcat(sir,v[i]);
        }
    }
    //for(i=1;i<=n;i++) cout<<v[i]<<' ';
    cout<<sir;
    return 0; }

```

3. Fișierul **bac.in** conține un șir de numere naturale distincte, din intervalul $[1, 10^9]$. Numerele din șir sunt separate prin câte un spațiu și cel puțin trei dintre ele au penultima cifră 2 și ultima cifră 0.

Se cere să se afișeze pe ecran cele mai mari trei numere din șir cu proprietatea că au penultima cifră 2 și ultima cifră 0. Numerele determinate sunt afișate în ordine crescătoare, separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă fișierul conține numerele 9731 50 112 20 8 16 8520 3 2520 1520

pe ecran se vor afișa, în această ordine, numerele: 1520 2520 8520

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

(8p.)

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

(2p.)

Vom reține în 3 variabile (m_1 , m_2 , m_3) cele 3 valori maxime $m_1 \leq m_2 \leq m_3$. Citim pe rând valorile din fișier; dacă ultimele cifre ale numărului curent sunt 2 și 0 atunci verificăm dacă poate face parte din cele trei valori maxime și reinitializăm valorile acestor variabile, după caz.

Programul foloseste putine variabile → programul va folosi pentru executie putina memorie.
Programul nu contine decat o instructiune repetitiva → programul necesita un timp mic pentru executie.

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

ifstream f("bac.in");

int main()
{
    unsigned long m1, m2, m3,x;

    m1=0;m2=0;m3=0;//initializam cele 3 valori maxime
    while(f>>x)
    {
        if(x%100==20)
        {
            if(x>m1) m1=x;
            if(x>m2) {m1=m2;m2=x;}
            if(x>m3) {m1=m2;m2=m3;m3=x;}
        }
    }
    cout<<m1<<' '<<m2<<' '<<m3;
    f.close();
    return 0;
}
```