

SUBIECTUL I

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Valoarea expresiei C/C++ alăturată este: (4p.) | 2018/3/22
- a. 30 b. 30.5758 c. 14784 d. 14798.7

2. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu [a] partea întreagă a numărului real a și cu a%b restul împărțirii numărului natural a la numărul natural nenul b.

- a) Scrieți ce se afișează dacă se citește numărul 9. 135*234*# (6p.)
- b) Scrieți cel mai mic număr natural care poate fi citit astfel încât, în urma executării algoritmului, să se afișeze o singură dată simbolul *. 6 (4p.)
- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat prima structură **cât timp...execută** cu o structură repetitivă de tip **pentru...execută**. (6p.)
- d) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```

citește n
    (număr natural)
x ← 1
cât timp x ≤ [n/3] execută
    y ← x+1
    cât timp y ≤ [n/3] execută
        z ← n-x-y
        dacă z%2=x%2 atunci
            scrie x,y,z,'*'
        ■
        y ← y+1
    ■
    x ← x+1
■
scrie '#'

```

c)

citește n

(număr natural)

~~x ← 1~~~~cât timp x ≤ [n/3] execută~~

pentru x=1,[n/3] executa

```

y ← x+1
cât timp y ≤ [n/3] execută
    z ← n-x-y
    dacă z%2=x%2 atunci
        scrie x,y,z,'*'
    ■
    y ← y+1
■

```

~~x ← x+1~~

scrie '#'

d)

```

#include <iostream>
using namespace std;
int n,x,y,z;

```

int main()

```

{
    cin >> n;
    x = 1;
    while(x <= n/3)
    {
        y = x+1;
        while(y <= n/3)
        {
            z = n-x-y;
            if(z%2 == x%2)
                cout << x << y << z << '*';
            y = y+1;
        }
        x = x+1;
    }
    cout << '#';
    return 0;
}

```

SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Indicați matricea de adiacență a unui graf orientat cu 4 vârfuri, numerotate de la 1 la 4, căruia îi poate aparține drumul 1, 3, 4, 3, 2. (4p.)
- a. $\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ b. $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ c. $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ d. $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$
2. Într-un arbore fiecare nod are cel mult 3 fii. Dacă 10 dintre nodurile sale au gradul egal cu 1, atunci numărul maxim de noduri cu gradul egal cu 4 este: (4p.)
- a. 3 b. 4 c. 5 d. 7

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Pentru o stație meteo se memorează, în variabila **m**, următoarele informații: luna și anul în care au fost făcute măsurători, precum și temperaturile medii înregistrate în **15** dintre zilele lunii respective.

Știind că expresiile C/C++ de mai jos au ca valori luna (un număr natural din intervalul **[1, 12]**) și anul în care au fost măsurate temperaturi (număr natural), respectiv prima temperatură medie înregistrată (un număr real), scrieți definiția unei structuri cu eticheta **meteo**, care permite memorarea informațiilor precizate, și declarați corespunzător variabila **m**.

```
struct {
    unsigned int luna, an;
    int temperatura[16];
}m;
```

(6p.)

4. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu **9** linii și **9** coloane, numerotate de la **0** la **8**, având inițial toate elementele nule.

Scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **a** să memoreze tabloul alăturat.

```
for(i=0; i<9; i++)
    for(j=0; j<9; j++)
        if(j<9-i-1 || j>9-i+1) a[i][j]=1;
        else a[i][j]=8;
```

(6p.)

1	1	1	1	1	1	1	8	8
1	1	1	1	1	1	8	8	8
1	1	1	1	1	8	8	8	1
1	1	1	1	8	8	8	1	1
1	1	1	8	8	8	1	1	1
1	1	8	8	8	1	1	1	1
1	8	8	8	1	1	1	1	1
8	8	8	1	1	1	1	1	1
8	8	1	1	1	1	1	1	1

5. O pereche de cuvinte, unul cu număr par de litere, iar celălalt cu număr impar de litere, se numește **descentrată** dacă se poate obține cuvântul cu număr par de litere din celălalt, prin duplicarea caracterului din mijlocul acestuia.

Exemplu: perechile **crezi** și **creezi**, respectiv **a** și **aa** sunt descentrate.

Un text are cel mult **100** de caractere, iar cuvintele sale 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 precizat mai sus și afișează pe ecran mesajul **DA**, dacă acesta conține cel puțin o pereche descentrată, sau mesajul **NU** în caz contrar.

Exemplu: dacă textul citit este
crezi ca poti sa creezi ceva original
se afișează pe ecran mesajul

DA

(10p.)

```
#include <iostream>
#include <string.h>
#include <stdlib.h>
```

```
using namespace std;
```

```
int main()
{
    char str[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, " ");
    }
}
```

```
/* for(i=1;i<=n;i++) cout<<v[i]<<' ';
cout<<endl; */
int gasit=0,jum,j;
```

```

i=1;
char aux[101];
while(i<=n && gasit==0)
{
    if(strlen(v[i])%2==0)
    {
        strcpy(aux,v[i]);
        jum=strlen(aux)/2;
        strcpy(aux+jum,aux+jum+1);
        j=1;
        if(v[i][jum]==v[i][jum-1])
            while(j<=n && gasit==0)
            {
                if(strcmp(aux,v[j])==0) gasit=1;
                j++;
            }
    }
    i++;
}

if(gasit==1) cout<<"DA";
else cout<<"NU";
return(0);
}

```

SUBIECTUL al III-lea

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

- | | |
|--|--|
| <p>1. Subprogramul f este definit alăturat. Valoarea f(2203, 2018) este:</p> <p style="text-align: right;">(4p.)</p> | <pre> int f(int x, int y) { if(x*y==0) return 0; if(x%2==y%2) return 1+10*f(x/10,y/10); return 10*f(x/10,y/10); } </pre> |
| <p>a. 1100 b. 11 c. 2 d. 0</p> | |

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Utilizând metoda backtracking, se generează toate posibilitățile de a forma șiraguri de câte **4** mărgelile de culori distincte din mulțimea {**roșu, albastru, roz, portocaliu, verde**}, astfel încât în fiecare șirag nu pot fi pe poziții alăturate mărgelile roșii și albastre. Două șiraguri sunt distincte dacă au cel puțin o mărgea de culoare diferită sau dacă ordinea culorilor mărgelilor este diferită. Primele cinci soluții generate sunt, în această ordine, (**roșu, roz, albastru, portocaliu**), (**roșu, roz, albastru, verde**), (**roșu, roz, portocaliu, albastru**), (**roșu, roz, portocaliu, verde**), (**roșu, roz, verde, albastru**). Scrieți cea de a șasea și cea de a șaptea soluție, în ordinea generării acestora. (6p.)
- (**roșu,roz,verde,portocaliu**)
(**roșu,portocaliu,albastru,roz**)
3. Se consideră subprogramul **radical**, cu trei parametri:
- **n**, prin care primește un număr natural ($n \in [1, 10^9]$);
 - **x** și **y**, prin care furnizează două numere naturale cu proprietatea că $\sqrt{n}$ poate fi scris sub forma $x \cdot \sqrt{y}$, unde **x** are o valoare maximă.
- Scrieți definiția completă a subprogramului.
- Exemplu:** pentru numărul **n=15000**, în urma apelului, **x=50** și **y=6**, iar pentru numărul **n=9**, în urma apelului, **x=3** și **y=1**. (10p.)

Impartim in factori primi si formam cele doua numere; numarul ce iese de sub radical se formeaza dintr-un produs al numerelor ce apar de cel putin doua ori in dezvoltarea in factori primi; la fiecare doua aparitii a unui factor prim il scoatem de sub radical.

```
void radical(unsigned long n, unsigned long &x, unsigned long &y)
{
    int d=2, nr, nn=n;
    x=1, y=1;
    while(n!=1)
    {
        nr=0;
        while(n%d==0)
        {
            n=n/d;
            nr++;
            if(nr==2){x=x*d;nr=0;}
        }
        d++;
    }
    y=nn/(x*x);
}
```

4. Se consideră un șir ai cărei termeni sunt numere naturale nenule, de o singură cifră. Numim **număr asociat** al acestui șir un număr natural format cu termenii șirului, în ordinea în care aceștia apar în șir.

Exemplu: numărul asociat șirului **1, 2, 5, 3, 2** este **12532**.

Fișierul text **bac.txt** conține numere naturale din intervalul **[1, 9]**: pe prima linie două numere, **x** și **y**, iar pe a doua linie un șir de cel puțin trei și cel mult 10^5 termeni. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu.

Se cere inserarea valorilor **x** și **y** în șirul aflat pe a doua linie fișierului, astfel încât numărul asociat șirului obținut să fie minim. Termenii șirului obținut se afișează pe ecran, separați prin câte un spațiu.

Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă fișierul **bac.txt** conține numerele

9 6

1 7 5

atunci, pentru că numerele asociate șirurilor care se pot obține sunt 96175, 69175, 61975, 61795, 61759, 91675, 19675, 16975, 16795, 16759, 91765, 19765, 17965, 17695, 17659, 91756, 19756, 17956, 17596, 17569, pe ecran se afișează șirul:

1 6 7 5 9

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

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

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

```
ifstream f("bac.txt");
```

```
int main()
{
    int x,y;
    f>>x>>y;
    int aux,a;
    if(x>y)
    {aux=x;x=y;y=aux;}
```

```
int pusx=0, pusy=0;
```

```
while(f>>a)
{
    if(a>x && pusx==0){cout<<x;pux=1;}
    if(a>y && pusy==0){cout<<y;pusy=1;}
    cout<<a;
}
if(pusx==0)cout<<x;
if(pusy==0)cout<<y;
f.close();
return 0;}
```

Deoarece ordinea cifrelor din fisier trebuie sa se pastreze, atunci vom afisa aceste cifre in ordinea citirii din fisier. Pentru citirea acestor cifre vom folosi o singura variabila in care vom citi valorile pe rand.

In cazul in care $x > y$ vom face o interschimbare a valorilor pentru cele doua variabile. Vom insera pe x imediat in fata primei valori mai mare ca x intalnite in fisier si vom afisa y in fata primei cifre din fisier mai mare ca el.

Daca cel putin una din valorile variabilelor x si y nu a fost afisata, atunci o/le vom afisa la sfarsit avand grija, eventual, la ordinea valorilor lui x si y , adica, $x \leq y$.

$pusx=0$ daca valoarea lui x nu s-a afisat si $pusx=1$ in caz contrar

$pusy=0$ daca valoarea lui xy nu s-a afisat si $pusy=1$ in caz contrar

Programul foloseste un numar mic de variabile, nu foloseste vectori deci spatiul ocupat de aceste variabile in memorie este mic.

Datorita faptului ca nu exista structuri repetitive imbricate, timpul de executie este un timp liniar, deci, timpul de executie al programului este mic.