

Sesiunea 16

Agenda

- Rezolvare varianta BAC Iunie 2022
- Rezolvare varianta BAC Sesiunea Speciala 2023

Rezolvare varianta BAC Iunie 2022

Subiectul I

1. ◦ Rezolvare:
 - Se respecta precedenta operatorilor si se acorda atentie casting-ului
- Raspuns corect: 10 -> **C**
2. ◦ Rezolvare:
 - Din punct de vedere sintatic, metoda corecta de atribuire este:
`valoare=2*x.cantitate*x.pret`
- Raspuns corect: **D**
3. ◦ Rezolvare:

Pentru usurinta rezolvarii notam sporturile astfel:

0 **1** **2** **3** **4**
tenis, fotbal, volei, handbal, baschet

Din enunt stim ca nu avem voie cu fotbal si baschet in
acelasi pachet si ca primele **5** solutii sunt:

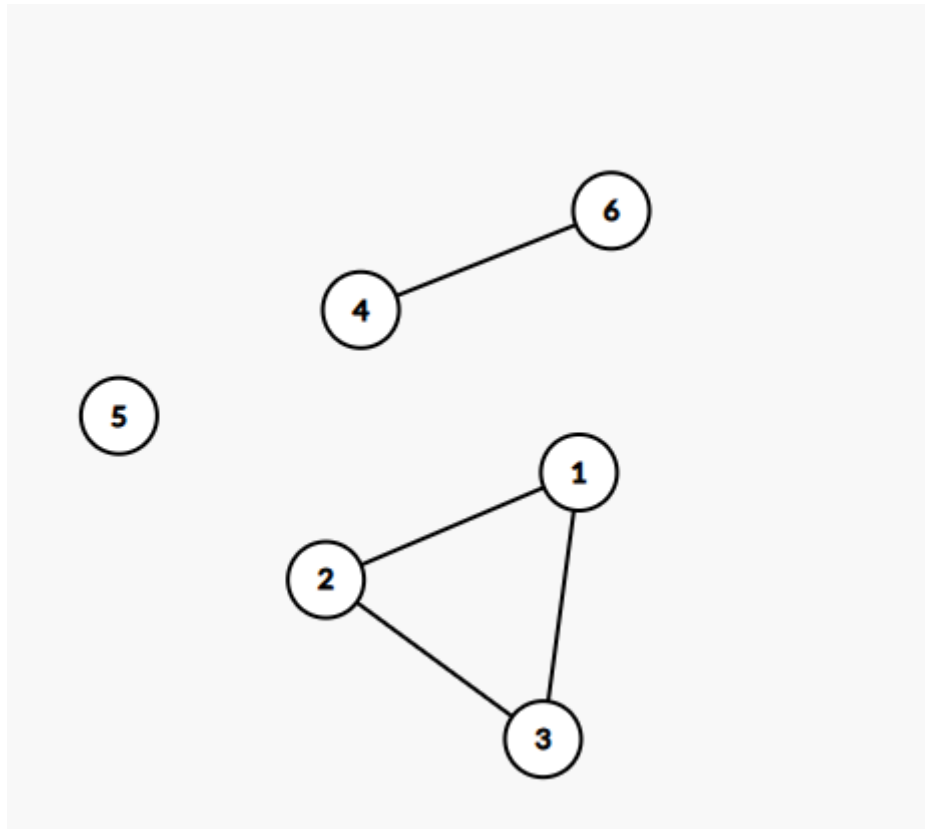
```
{tenis, fotbal}  
{tenis, fotbal, volei}  
{tenis, fotbal, handbal}  
{tenis, volei}
```

Adica:

```
0 1  
0 1 2  
0 1 3  
0 2  
0 2 3  
0 2 4  
1 2  
1 2 3  
1 3 [Fotbal Handbal]  
2 3 [volei handbal ]
```

- Raspuns corect: **A**
4. ◦ Rezolvare:

- Conform enuntului avem graful de mai jos:



Deci avem 3

componente conexe

- Raspuns corect: B

5. ◦ Rezolvare:

Pentru a rezolva exercitiul trebuie sa generam arborele pentru fiecare dintr cazuri, din vectorul de tati obtinut prin inlocuirea lui x si y

- a

```

x=11 y=12
2 4 2 0 4 4 x 6 x x y y DEVINE
2 4 2 0 4 4 11 6 11 11 12 12
Astfel avem urmatorul vector de tati:
1 2 3 4 5 6 7 8 9 10 11 12
2 4 2 0 4 4 11 6 11 11 12 12
unde:
4: radacina
4 parinte pentru: 2 5 6
2 parinte pentru: 1 3
6 parinte pentru: 8
11 parinte pentru: 7 9 10
12 parinte pentru: 11 si 12
*Deoarece avem un ciclu, acesta nu poate fi un arbore
  
```

o b

```
x=8 y=3
2 4 2 0 4 4 x 6 x x y y DEVINE
2 4 2 0 4 4 8 6 8 8 3 3
```

Astfel avem urmatorul vector de tati:

```
1 2 3 4 5 6 7 8 9 10 11 12
2 4 2 0 4 4 8 6 8 8 3 3
```

unde:

4: radacina

4 parinte pentru: 2 5 6

2 parinte pentru: 1 3

8 parinte pentru: 7 9 10

6 parinte pentru: 8

3 parinte pentru: 11 12

Astfel rezulta urmatorul graf:

! [Poza graf] (imagini/s1e5-b.png)

Din poza rezulta ca nu este graful pe care il cautam deoarece desi avem 5 noduri frunza pe ultimul nivel, nu avem un nod numerotat cu numarul sau de fii.

o c

```
x=3 y=8
2 4 2 0 4 4 x 6 x x y y DEVINE
2 4 2 0 4 4 3 6 3 3 8 8
```

Astfel avem urmatorul vector de tati:

```
1 2 3 4 5 6 7 8 9 10 11 12
2 4 2 0 4 4 3 6 3 3 8 8
```

Unde:

4: radacina

4 parinte pentru: 2 5 6

2 parinte pentru: 1 3

3 parinte pentru: 7 9 10

6 parinte pentru: 8

8 parinte pentru 11 si 12

Astfel rezulta urmatorul graf:

! [Poza graf] (imagini/s1e5-c.png)

De unde vedem ca ambele conditii sunt respectate, avem 5 noduri frunza pe ultimul nivel dar si un nod ce are o valoare egala cu numarul de fii (3)

o Raspuns corect: C

1. ◦ a.

```

n = 56
i = 2
k = 0
cat timp 56 > 2 executa
    cat timp 56 % 2 == 0 executa
        k = k+1 = 1
        n = n/i => n = 56/2 => 28
    cat timp 28 % 2 == 0 executa
        k = k+1 = 2
        n = n/i => n = 28/2 => 14
    cat timp 14 % 2 == 0 executa
        k = k+1 = 3
        n = n/i => n = 14/2 => 7
    cat timp 7%2 == 0 => false

    Daca i == 2 FALSE => i => 4
cat timp 7 >= 3 executa
    cat timp 7 % 3 executa => false
    daca i == 2 => false=> i => 5
cat timp 7 >= 5 executa
    cat timp 7 % 5 executa => false
    daca i == 2=> false => i=>7
cat timp 7 >= 7 executa
    cat timp 7 % 7 executa
        k = k+1 => 4
        n = 0
    daca i == 2=> false => i=>9
scrie k => scrie 4

```

◦ b

- Algoritmul afiseaza numarul de divizori ai unui numar, deci orice numar prim din acel interval satisface conditia:

```

11
13
17
19
23
29
31
37
41
43
47
53
59
61
67

```

```

71
73
79
83
89
97

```

-c ``c++ #include using namespace std;

```

int main()
{
    int n;
    cin >> n;
    int i = 2, k = 0;
    while (n >= i) {
        while (n % i == 0) {
            k = k+1;
            n = n/i;
        }
        if (i == 2) {
            i = i+1;
        } else {
            i = i+2;
        }
    }
    cout << k;
    return 0;
}
...

```

o d

```

citește n
(număr natural)
i<-2; k<-0
┌cât timp n≥i execută
│daca n%i = 0 atunci
││ execută
││ │ k<-k+1
││ │ n<-[n/i]
││ cat timp n%i = 0
│└─
│┌daca i=2 atunci i<-i+1
││altfel i<-i+2
│└─
└─
scrie k

```

2. ◦ Rezolvare
 - Uitandu-ne la intervalul dat si la functia specificata, observam, ca trebuie sa ajungem la un caz (de oprire) in care $a \leq b$ si a sa aibe valoarea 2022
 - Prin urmare 2022, 2023,2024,2025. Pentru oricare din valori se va intoarce valoarea 2022
3. ◦ Rezolvare

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

int main()
{
    char s[51] = "Ana Popescu";
    char id[51];
    id[0] = '\0';

    char *nume = strtok(s, " ");
    nume = strtok(NULL, " ");
    strcat(id, nume);
    strcat(id, "2022");
    cout << id;

    return 0;
}
```

Subiectul III

1. ◦ Rezolvare:

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

int secventa(int n);

int main()
{
    int n = 202233228;
    cout << secventa(n);
    return 0;
}

int secventa(int n) {
    int p = 1;
    int rezultat = 0;
```

```

int primaCifraRezultat=0;
while (n > 0) {
    int ultimaCifra = n % 10;
    n = n /10;
    if (ultimaCifra == 2 && primaCifraRezultat == 2) {
        p = p/10; // ne mutam cu o pozitie in spate
        rezultat = rezultat % p; // si taiem prima cifra
        rezultat = 20 * p + rezultat;
        primaCifraRezultat = 2;
        p = p * 10; // punem inapoi pozitia taiata
    } else {
        rezultat = ultimaCifra * p + rezultat;
        primaCifraRezultat = rezultat / p;
    }

    p = p * 10;
}

return rezultat;
}

```

2.
 - Rezolvare:
 - Solutie:

```

#include <iostream>

int calculeazaMinimum(int a, int b, int c, int d);
int calculeazaMinimum(int a, int b, int c);
int calculeazaMinimum(int a, int b);

using namespace std;

int main() {
    int m, n;
    cin >> m >> n;
    int matrice[m][n];

    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            cin >> matrice [i][j];
        }
    }

    int nisipAaugat = 0;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            int minValue;
            int parcelaNord = matrice[i-1][j];
            int parcelaEst = matrice[i][j + 1];
            int parcelaSud = matrice[i + 1][j];

```

```
int parcelaVest = matrice[i][j-1];
// daca suntem pe prima linie
if (i == 0) {
    // si suntem si pe prima coloana
    if (j == 0) {
        // atunci avem vecini doar in est si in
sud (sau dreapta si jos)
        minValue = calculeazaMinimum(parcelaEst,
parcelaSud);
    } else if (j == n-1) {
        // dar daca suntem ultima coloana din
prima linie
        // avem vecini doar in sud si vest (sau
dreapta si stanga)
        minValue = calculeazaMinimum(parcelaSud,
parcelaVest);
    } else {
        // altfel o sa avem vecini in sud, vest
si est (sau jos, dreapta si stanga)
        minValue = calculeazaMinimum(parcelaSud,
parcelaVest, parcelaEst);
    }
} else if (i == m-1) {
    // insa daca suntem pe ultima linie
    if (j == 0) {
        //si pe prima coloana
        //atunci avem vecini doar in nord si in
est (sau sus si in dreapta)
        minValue = calculeazaMinimum(parcelaNord,
parcelaEst);
    } else if (j == n-1) {
        // insa daca suntem pe ultima coloana
        // avem vecini in nord si in vest (sau
sus si stanga )
        minValue = calculeazaMinimum(parcelaNord,
parcelaVest);
    } else {
        // altfel o sa avem vecini atat in nord
cat si in vest si est
        // sau sus, dreapta, stanga
        minValue = calculeazaMinimum(parcelaNord,
parcelaVest, parcelaEst);
    }
} else {
    // altfel, daca nu suntem nici pe prima nici
pe ultima trebuie din nou sa verificam pe ce coloana suntem

    //daca suntem pe prima coloana
    if (j == 0) {
        // atunci avem vecini in nord, est si sud
        minValue = calculeazaMinimum(parcelaNord,
parcelaEst, parcelaSud);
    } else if (j == n-1) {
        // dar daca suntem pe ultima coloana
```



```

        // avem vecini in nord, sud si vest
        minValue = calculeazaMinimum(parcelaNord,
parcelaSud, parcelaVest);
    } else {
        // altfel suntem pe o pozitie in care
        avem vecini in toate partile
        minValue = calculeazaMinimum(parcelaNord,
parcelaEst, parcelaVest, parcelaSud);
    }
}

    if (matrice[i][j] < minValue) {
        int diferenta = minValue - matrice[i][j];
        matrice[i][j] = diferenta;
        nisipAdaugat += diferenta;
    }
}

cout << nisipAdaugat;
}

int calculeazaMinimum(int a, int b) {
    if (a < b) {
        return a;
    } else {
        return b;
    }
}

int calculeazaMinimum(int a, int b, int c) {
    int minBC = calculeazaMinimum(b, c);
    return calculeazaMinimum(a, minBC);
}

int calculeazaMinimum(int a, int b, int c, int d) {
    int min = a;
    if (b < min) min = b;
    if (c < min) min = c;
    if (d < min) min = d;
    return min;
}

```

3. Rezolvare:

- a

Vom implementa un algoritm care va citii numerele din fisier, unul cate unul, si la fiecare numar verificam daca este in intervalul specificat, caz in care incrementam un contor. De asemenea, o sa salvam de fiecare data ultimul

numar pentru care am incrementat contorul pentru ca sa stim sa ignoram urmatoarele numere care sunt egale cu acesta deoarece trebuie sa le numarm doar pe cele unice.

Algoritmul este eficient din punct de vedere al timpului de executie deoarece parcurgem o singura data fisierul, si in acelasi timp este eficient din punct de vedere al memoriei deoarece din maximul de 1 milion de numere, noi in memorie tinem doar 2 numere.

■ b

```
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ifstream fin("bac.txt");
    int limitaStanga, limitaDreapta;
    fin >> limitaStanga >> limitaDreapta;

    int contor = 0;
    int numar, ultimulNumarCitit;
    fin >> numar;
    ultimulNumarCitit = numar;
    if (numar >= limitaStanga && numar <= limitaDreapta)
    {
        contor++;
    }
    while (fin >> numar) {
        if (numar == ultimulNumarCitit) {
            continue;
        } else if (numar >= limitaStanga && numar <=
limitaDreapta) {
            contor++;
            ultimulNumarCitit = numar;
        }
    }

    cout << contor;

    fin.close();
    return 0;
}
```

Rezolvare varianta BAC Sesiunea Speciala 2023

Subiectul I

1. ◦ Rezolvare:

- Conform cerintei doar daca ambele sunt pare trebuie sa obtinem 1
- a. Este invalid deoarece obtinem 1 si daca unul este par si unul impar
- b. Este invalid deoarece obtinem 1 si daca, de exemplu, ambele sunt impare (e.g $m = 7$, $n = 3$)
- c. Este adevarat doar daca ambele sunt pare, rezulta ca c este raspunsul
- d. Este invalid pentru ca obtinem 1 daca ambele sunt impare.

◦ Raspuns corect: c

2. ◦ Rezolvare:

$$\begin{aligned}
 f(35, 3) &= \\
 &= 2 + 10 * f(11, 3) = \\
 &= 2 + 10 * f(3, 3) = \\
 &= 0 + 10 * f(1, 3) \\
 &= 1 + 10 * f(0, 3) = 1 \\
 &= 10 \\
 &= 2 + 10 * 10 = 102 \\
 &= 2 + 10 * 102 = 1022
 \end{aligned}$$

◦ Raspuns corect: a

3. ◦ Rezolvare:

formula corecta ar fi: $v = p.lg * p.lt * p.h$
 Datorita sintaxei incorecte, se elimina variantele a,b,c

◦ Raspuns corect: d

4. ◦ Rezolvare:

Pentru usurinta notam astfel:
 A M U R G
 0 1 2 3 4

Primele 5 solutii generate sunt:
 URAMG
 URAGM
 URMAG
 URMGA
 URGAM
 Adica:
 2 3 0 1 4
 2 3 0 4 1

```

2 3 1 0 4
2 3 1 4 0
2 3 4 0 1

```

Trebuie sa zicem ce urmeaza dupa:

GURMA => 4 2 3 1 0 si stim ca litera R precede literele A si M adica 3 precede 0 si 1

Observand felul in care se genereaza cele 5 solutii initiale, urmatoarea ar fi 4 3 0 1 2 adica: GRAMU

- Raspuns corect: b

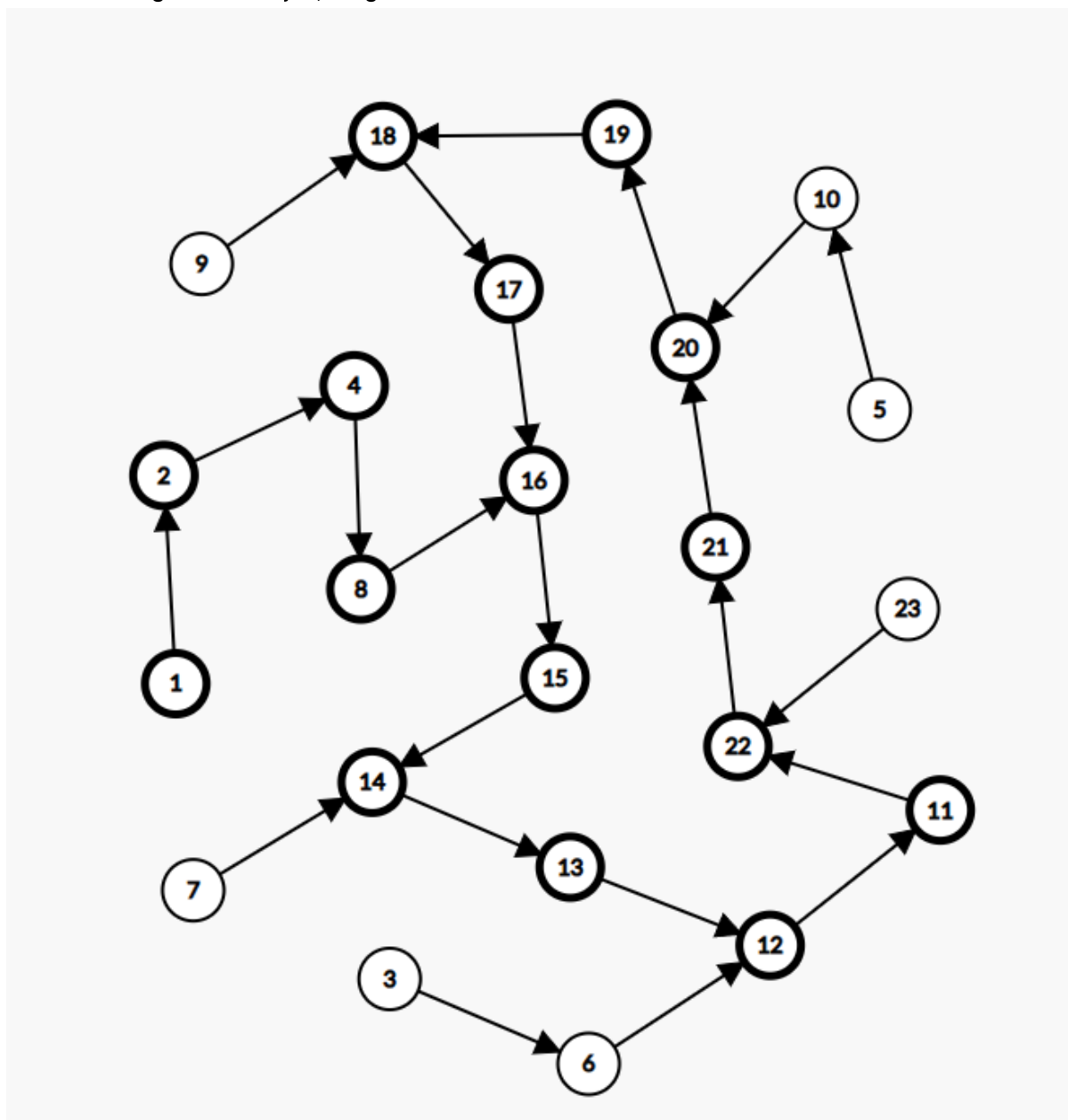
5. ◦ Rezolvare:

- Drum elementar: Un lanț (drum) se numește elementar dacă în el nu se repetă noduri

- Avand 23 de noduri si conditia din enunt inseamna ca avem arcele:

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

- Raspuns corect: **C**



- Raspuns corect: **C**

1. o a

- ```
n = 3
i = 1
 j = 1
 j <= i adevarat => scriem 2
 j = 2
 j <= i fals => scriem 3
 j = 3
 j <= i fals => scriem 3
i = 2
 j = 1
 j <= i adevarat => scriem 2
```

```

 j = 2
 j <= i adevarat => scriem 2
 j = 3
 j <= i adevarat => scriem 3
 i = 3
 j = 1
 j <= i adevarat => scriem 2
 j = 2
 j <= i adevarat => scriem 2
 j = 3
 j <= i adevarat => scriem 2
Raspuns corect: 233223222

```

o b

■ Rezolvare:

Daca ar fi sa facem codul sub forma unei matrici patratiche, s-ar observa ca afisam un 2 pentru toate pozitiile care sunt fie pe diagonala principala sau sub ea. Si aici e o formula, numarul de elemente sub diagonala principala este egala cu suma primelor n numere naturale consecutive adica  $(n * (n+1))/2$ . Deci de exemplu daca:

```

n = 3 => 6 de 2
n = 4 => 10 de 2
n = 5 => 15 de 2
n = 6 => 21 de 2
n = 7 => 28 de 2
n = 8 => 36 de 2
n = 9 => 45 de 2
n = 10 => 55 de 2

```

Rezulta ca putem alege oricare 2 numere din [6,9]

o c

```

#include <iostream>

using namespace std;

int main()
{
 int n;
 cin >> n;
 for (int i = 1; i<=n;i++) {
 for (int j = 1; j <=n; j++) {
 if (j <= i) {
 cout << 2;
 } else {
 cout << 3;
 }
 }
 }
}

```

```

 }
 }
 return 0;
}

```

o d

```

citește n
(număr natural nenul)
pentru i<-1,n execută
| j <- 1
| cat timp j<= n execută
| | dacă j≤i atunci
| | | scrie 2
| | | altfel
| | | scrie 3
| | ■
| | j <- j+1
| | ■
| ■
■

```

2. o Rezolvare:

Din enunt avem vectorul de tati:

```

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

```

Din care rezulta:

1 -> Radacina

1 parinte pentru -> 2, 3, 8

2 parinte pentru -> 4, 5

4 parinte pentru -> 6

5 parinte pentru -> 7

Rezulta ca descendentii nodului 2 sunt: 4, 5, 6, 7

3. o Rezolvare:

```

#include <iostream>
#include <cstring>

using namespace std;

int main()
{
 char s[21] = "elemente";
 int k;

```

```

char a[21];
for (k = strlen(s)-1; k>= 0; k--) {
 if (s[k] == s[0]) {
 int i = 0;
 for (i =0; i <=k; i++) {
 a[i] = s[i];
 }
 a[i] = '\0';
 cout << a <<" ";
 }
}

return 0;
}

```

### Subiectul III

1.     ◦ Rezolvare:

```

#include <iostream>

using namespace std;

void Putere (int n, int &x, int &p);

int main()
{
 int n=12, x, p;
 Putere(n, x, p);
 cout << x <<" " << p;
 return 0;
}

void Putere (int n, int &x, int &p) {
 x = 2;
 p = 0;
 int copieN = n;

 while (copieN > 1) {
 while (copieN % x == 0) {
 copieN = copieN / x;
 p++;
 }
 if (copieN > 1) {
 copieN = n;
 x++;
 p = 0;
 }
 }
}

```



## 2.   ◦ Rezolvare:

```
#include <iostream>

using namespace std;

int main()
{
 int n;
 cin >> n;
 int matrice[n][n];
 for (int i = 0; i < n; i++) {
 cin >> matrice[0][i];
 }

 for (int i = 1; i < n; i++) {
 for (int j = 0; j < n; j++) {
 if (j > n-1-i) {
 matrice[i][j] = -1;
 } else {
 int suma = matrice[i-1][j] + matrice[i-1]
[j+1];

 int cod = suma % 100;
 matrice[i][j] = cod;
 }
 }
 }

 for (int i = 0; i < n; i++) {
 for (int j = 0; j < n; j++) {
 cout << matrice[i][j] << " ";
 }
 cout << endl;
 }
 return 0;
}
```

## 3.   ◦ Rezolvare:

## ■ a:

Algoritmul este eficient din punct de vedere al timpului de executie deoarece se efectueaza o singura citire a fisierului. In acelasi timp, algoritmul este eficient din punct de vedere al memoriei deoarece nu utilizam alte structuri de date pentru a memora numerele din fisier.

## ■ b:

```
#include <iostream>
#include <fstream>

using namespace std;

int main()
{
 ifstream fin("bac.in");
 int min, max;
 int p1 = 0, p2 = 0;
 int k, numar;
 fin >> numar;
 min = max = numar;
 k = 1;
 while (fin >> numar) {
 k++;
 if (numar < min) {
 min = numar;
 } else if (numar > max) {
 max= numar;
 } else {
 if (numar >= min && numar <= max && p1 == 0)
{
 p1 = k;
 } else {
 p2 = k;
 }
 }
 }

 if (p1 != 0 && p2 != 0) {
 cout << p1 <<" " << p2;
 } else {
 cout << "nu exista";
 }

 fin.close();
 return 0;
}
```