

Sesiunea 12

Agenda

- Rezolvare Model Propus 2026
- Ce putem face mai bine
- Exercitii pentru acasa

Rezolvare Model Propus 2026

Subiectul I

1. Rezolvare

- Observam ca avem de a face cu numere intregi deci orice rezultat care are virgula este trunchiat:

$$\begin{aligned} 20/25 &= 0 \\ 0 * 20 &= 0 \\ 0 / 2 &= 0 \end{aligned}$$

- Raspuns corect: A

2. Rezolvare

```
int x[] = {2,0,2,6,8};
f(0,4,x) =
    return f(0,2,v) + f(3,4,v)

f(0,2,v) =
    return f(0, 1,v) + f(2, 2, v)

f(0,1,v) =
    return f(0,0, v) + f(1, 1, v) = 0 + 0 =0
f(2,2,v) = 0

f(3,4,v) = f(3,3,v) + f(4,4,v) = 1 + 1 = 2
--- egal 2
```

- Raspuns corect B

3. Rezolvare

3	0	1	2
	4		

{jenga (motricitate), kendama (motricitate), lego (creativitate), şah

(strategie), scrabble (vocabular)}

(jenga, lego, şah),
(jenga, lego, scrabble),
(jenga, şah, lego),
(jenga, şah, scrabble),
(jenga, scrabble, lego)

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

3 2 4
3 4 2

- Raspuns corect: A

4. Rezolvare

```
- a => Valid
- b => Invalid dpdv sintactic
- c => Invalid dpdv sintactic
- d => Invalid dpdv sintactic
```

- Raspuns corect A

5. Rezolvare

```
- Dupa ce desenam graful observam ca ne lipseste latura [4,5]
- Lucrul asta ne face sa ramanem la variantele a si c
- Daca alegem a, observam ca invalidam cerinta, anume lantul elementar
cu lungimea maxima nu va mai fi cel dar
- Deci pentru a pastra lungimea maxima alegem `C` => [2,3], [4,5]
```

- Raspuns corect: C

Subiectul II

1. ○ a

```
m = 27, n = 38
reepeta
x = 27
y = 38
```

```
n = 37
repeta
    daca x > y fals
        altfel y = 38-27=11
    pana cand y = 0
repeta
    daca x > y => x = 16
    pana cand y = 0
repeta
    daca x > y => x = 5
    pana cand y = 0
repeta
    daca x > y fALS
        atunci y = 6
    pana cand y = 0
repeta
    daca x > y fals
        altfel => y = 1
    pana cand y = 0
repeta
    daca x > y => x = 4
    pana cand y = 0
repeta
    daca x > y => x = 3
    pana caND y = 0
repeta
    daca x > y => x = 2
    pana cand y === 0
repeta
    daca x > y => x = 1
    pana cand y == 0
repeta
    daca x > y
        altfel => y = 0
    pana cand y == 0
pana cand x != 1
repeta
    x = 27,
    y = 37
    n = 36
repeta
    daca x > y fals
        altfel => y = 10
    pana cand y == 0
repeta
    daca x > y true => x = 17
    pana cand y == 0
repeta
    daca x > y true => x = 7
    pana cand y == 0
repeta
    daca x > y fals
        altfel y = 3
    pana cand y == 0
```

```

repeta
    daca x > y true => x = 4
    pana cand y == 0
    repeta
        daca x > y true => x = 1
        pana cand y == 0
        repeta
            daca x > y fals
            altfel y = 2
            pana cand y == 0
            repeta
                daca x > y fals
                altfel y = 1
                pana cand y == 0
                repeta
                    daca x > y
                    altfel y = 0
                    pana cand y == 0
                    pana cand x!=1
repeta
    x = 27
    y = 36
    n = 35
    repeta
        daca x > y fals
        altfel y = 9
        pana cand y == 0
    repeta
        daca x > y true => x = 18
        pana cand y == 0
    repeta
        daca x > y true => x = 9
        pana cand y == 0
    repeta
        daca x > y fals
        y = 0
        pana cand y == 0
    pana cand x != 1 (true => x = 9)

scrie n +1 => scrie 36

```

◦ b

algoritmul calculeaza CMMDC folosind Algoritmul lui Euclid prin scadere repetata (https://en.wikipedia.org/wiki/Euclidean_algorithm)

- dar aplică niște artificii.
- Mai precis, cauta primul n al carui $\text{cmmdc}(m,n)$ să fie diferit de $1 \rightarrow$
- Dacă ar fi să incepem cu $n = 15$
 - $\text{gcd}(5, 15) \Rightarrow 5$ se va afisa 15 nu e ok deoarece vrem să afisam 10
 - Setăm $n = 14$ și calculăm repetitiv:

```

    - gcd(5, 14) -> 1 merem mai departe
    - gcd(5, 13) -> 1 merem mai departe
    - gcd(5, 12) -> 1 merem mai departe
    - gcd(5, 11) -> 1 merem mai departe
    - gcd(5, 10) -> 5 e ok => afisam n +1 unde n este 9
deoarece la inceputul iteratie scadem 1 din el, si afisam 10 (9+1)
[10,14]

```

o c

```

#include <iostream>

using namespace std;
int main(){
    int m, n;
    cin >> m >> n;
    int x,y;
    do {
        x = m;
        y = n;
        n = n -1;
        do {
            if ( x > y) {
                x = x - y;
            } else {
                y = y - x;
            }
        } while (y != 0);
    } while (x == 1);
    cout << n+1;
    return 0;
}

```

o d

```

citește m,n
(numere naturale, 1<m<n)
repetă
| x<-m; y<-n; n<-n-1
| cat timp y != 0 execută
| | dacă x>y atunci x<-x-y
| | altfel y<-y-x
| |
| ┌
|
└ până când x≠1
scrie n+1

```

- Atentie
 - Daca are 2 frati inseamna ca parintele lui trebuie sa aibe 3 copii
- Posibile variante: 2,6

3. Rezolvare

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

int main(){
    char tI[21], pN[21], tL[21];
    cin.getline(tI, 21);
    cin.getline(pN, 21);
    strcpy(tL, pN);
    strcat(tL, strchr(tI, ')')+1);
    cout << tL << endl;
    return 0;
}
```

Subiectul III

1. Rezolvare

```
#include <iostream>

using namespace std;

int Plus(int);

int main() {
    cout << Plus(202535250);
}

int Plus(int n) {
    int rezultat = 0;
    int p = 1;
    while (n) {
        int ultimaCifra = n % 10;
        n = n / 10;
        if (ultimaCifra == 5 && n % 10 == 2) {
            rezultat = 6 * p + rezultat;
            p = p * 10;
            rezultat = 2 * p + rezultat;
            p = p * 10;
            n = n / 10;
        } else {
            rezultat = ultimaCifra * p + rezultat;
            p = p * 10;
        }
    }
}
```

```

    }
    return rezultat;
}

```

2. Rezolvare

```

#include <iostream>

using namespace std;

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

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

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - 1; j++) {
            cout << matrice[i][j] << " ";
        }
        cout << endl;
    }
}

```

- Notă: rezolvarea de mai sus presupune suprascrierea elementelor prin mutarea mai la stanga cu o celula a tuturor elementelor din dreapta diagonalei secundare pana la elementul fix de pe diagonala secundara. Iar la afisare, se observă că **j** merge pana la **n-1** deoarece noi am suprascris o valoarea
 - O alta soluție ar fi fost construcția unei matrice care să aibă același număr de linii ca cea initială dar cu $n-1$ coloane și în care să copiem elementele mai puțin cele de pe diagonala secundara.

3. Rezolvare a.

Algoritmul este eficient deoarece nu calculează efectiv valoarea lui $n!$, care ar fi foarte mare, ci determină doar de câte ori apar factorii primi 2 și 13 în descompunerea acestuia. Pentru aceasta, se folosesc împărțiri repetitive ale lui n cu 2, respectiv cu 13, operații simple și rapide. Numărul de pași este mic, iar memoria folosită este minimă, deoarece se utilizează doar câteva variabile întregi. În final, se alege cel mai mic dintre cei doi exponenti, deoarece fiecare factor 26 este format dintr-un 2 și un 13.

b.

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

using namespace std;

long long numaraFactori(long long n, int prim);

int main() {
    long long n;
    cin >> n;
    long long numara2 = numaraFactori(n, 2);
    long long numara13 = numaraFactori(n, 13);
    long long rezultat = min(numara2, numara13);

    ofstream fout("bac.txt");
    fout << rezultat;

    fout.close();
}

long long numaraFactori(long long n, int prim) {
    long long count = 0;
    long long p = prim;
    while (p <= n) {
        count += n / p;
        // prevenim overflow adica sa nu avem p * prime > n
        if (p > n / prim) {
            break;
        }
        p *= prim;
    }
    return count;
}
```

Ce putem face mai bine

- Aici o sa gasesti chestiile pe care le-am observat si pe care le putem imbunatatii astfel incat sa iti fie mai usor la examen
1. Nu este gresit sa copiezi cod pe net sau sa folosesti cod generat de chat gpt dar este gresit daca faci asta fara sa il intelegi si fara sa incerci tu singur sa il scrii.
 2. Chat GPT sau orice al tool similar este util atunci cand il folosim nu pentru a crea de la 0 ci atunci cand ii aratam ce a facut si ne explica de ce nu merge si/sau cum putem sa avem o rezolvare mai clara
 3. Chat GPT sau orice alt tool similar poate sa halucineze de aceea nu trebuie sa luam 100% de bun ce scrie acolo
 4. Atunci cand nu intelegi un concept pe care l-am discutat in trecut, te poti uita in pdf-urile pe care le-am trimis. Exact asta e motivul pentru care iti trimit la finalul sedintei un fisier cu ceea ce am rezolvat. Este foarte util ca tu sa mai treci cel putin o data prin el singur si sa vezi ca ai intelese, iar daca nu, sa putem relua acel topic.
 5. Orice varianta rezolvata, fa-o mai intai pe foaie dupa care verifica-te pe PC dar nu in notepad.. ci in Clion/Codeblocks altfel nu ai idee daca l-ai facut ok sau nu.
 6. Nu trata superficial problemele, citeste cu atentie enuntul
 7. Fa cat mai multe exercitii altfel uiti si ce ai facut inainte.. dupa cum ai vazut, desi am stat si am discutat despre functii cam la fiecare sedinta, conceptelete de apel de functie si parametrii iti pareau putin nefamiliale.

Stiu ca pare ca is multe de facut insa nu e asa, putem sa recuperam daca iti dai interesul ca la fiecare exercitiu sa il faci cat de bine poti tu.Si sa faci cat mai multe exercitii. Lucru valabil pentru toate materiile de bac.

Spor!

Exercitii pentru acasa

1. Rezolvare varianta Bac August 2024 =>

https://www.pbinfo.ro/resurse/9dc152/examene/2025/E_D_Informatica_2025_sp_MI_C_var_04_LRO.pdf

2. Exercitii recapitative functii prelucrare siruri.

Nota: aici gasesti functiile de prelucrare siruri cu tot cu exemple: <https://cplusplus.com/reference/cstring/> si alegi functia de care ai nevoie

◊ Problema 1

Scrie un program C++ care sa copieze un sir de caractere in alt sir, folosind functia `strcpy`.

Exemplu date de intrare:

```
informatica
```

Exemplu date de ieșire:

```
Copia sirului este: informatica
```

◊ Problema 2

Scrie un program C++ care să concateneze două siruri de caractere, folosind funcția `strcat`.

Exemplu date de intrare:

```
info  
matica
```

Exemplu date de ieșire:

```
Rezultatul concatenarii: informatica
```

◊ Problema 3

Scrie un program C++ care să compare două siruri și să afișeze dacă sunt egale sau care este mai mare lexicografic, folosind funcția `strcmp`.

Exemplu date de intrare:

```
ana  
anca
```

Exemplu date de ieșire:

```
Primul sir este mai mic lexicografic decat al doilea.
```

◊ Problema 4

Scrie un program C++ care să copieze doar primele `n` caractere dintr-un sir în altul, folosind funcția `strncpy`.

Exemplu date de intrare:

```
informatica 5
```

Exemplu date de ieșire:

```
Primele 5 caractere copiate: infor
```

◊ Problema 5

Scrie un program C++ care să concateneze doar primele **n** caractere din al doilea sir la primul sir, folosind funcția **strncat**.

Exemplu date de intrare:

```
info matica 4
```

Exemplu date de ieșire:

```
Rezultatul concatenarii partiale: infomat
```

◊ Problema 6

Scrie un program C++ care să verifice dacă un caracter dat se află într-un sir de caractere, folosind funcția **strchr**.

Exemplu date de intrare:

```
informatica a
```

Exemplu date de ieșire:

```
Caracterul 'a' se gaseste in sir.
```

◊ Problema 7

Scrie un program C++ care să determine de câte ori apare un anumit caracter într-un sir, folosind funcția **strchr**.

Exemplu date de intrare:

informatica a

Exemplu date de ieșire:

Caracterul 'a' apare de 2 ori.

◊ Problema 8

Scrie un program C++ care să verifice dacă un sir este subșir al altui sir, folosind funcția `strstr`.

Exemplu date de intrare:

informatica mat

Exemplu date de ieșire:

Al doilea sir este subșir al primului.

◊ Problema 9

Scrie un program C++ care să steargă un subșir dintr-un sir dat, dacă acesta apare, folosind funcțiile `strstr`, `strcpy` și `strcat`.

Exemplu date de intrare:

informatica mat

Exemplu date de ieșire:

Sirul după stergerea subșirului: inforica

◊ Problema 10

Scrie un program C++ care să înlocuiască toate aparițiile unui subșir dintr-un sir cu alt subșir, folosind funcțiile `strstr`, `strcpy` și `strcat`.

Exemplu date de intrare:

informatica mat log

Exemplu date de ieșire:

Sirul după înlocuire: infologica

-
-
3. Pentru problemele de mai jos, nu este nevoie să scrii efectiv codul C++ ci doar să identifici pasii pe care îi ai de facut împreună cu ce functii te-ar putea ajuta să faci asta

◊ Exercitiul 1

Scrie un program C++ care citește un text format din mai multe cuvinte și formează un nou sir în care toate cuvintele sunt concatenate într-un singur sir,
dar fiecare cuvânt este prescurtat la primele 3 caractere.

Exemplu date de intrare:

informatica este frumoasa

Exemplu date de ieșire:

infestfru

◊ Exercitiul 2

Scrie un program C++ care citește două propoziții și verifică dacă au vreun cuvânt comun.
Dacă există cuvinte comune, acestea se vor afișa separate printr-un spațiu,
iar dacă nu există, se va afișa mesajul „Nu există cuvinte comune.”

Exemplu date de intrare:

ana are mere
are pere si mere

Exemplu date de ieșire:

are mere

◊ Exercitiul 3

Scrie un program C++ care citește un text și creează un nou sir care conține doar cuvintele din text ce au lungimea mai mare decât 4 caractere.

Cuvintele rezultate se vor separa printr-un singur spațiu.

Exemplu date de intrare:

```
azi invatam programare in c++
```

Exemplu date de ieșire:

```
invatam programare
```

◊ Exercitiul 4

Scrie un program C++ care citește un sir de caractere și formează un nou sir ce conține ultimele 2 caractere din fiecare cuvânt.

Cuvintele sunt separate printr-un spațiu.

Exemplu date de intrare:

```
mere pere prune
```

Exemplu date de ieșire:

```
re re ne
```

◊ Exercitiul 5

Scrie un program C++ care citește o propoziție și construiește un nou sir în care cuvintele sunt aranjate în ordine alfabetică.

Exemplu date de intrare:

```
ion are mere si pere
```

Exemplu date de ieșire:

are ion mere pere si

◊ Exercitiul 6

Scrie un program C++ care citește o propoziție și elimină toate cuvintele care conțin o literă dată.

Exemplu date de intrare:

```
ana are mere rosii  
a
```

Exemplu date de ieșire:

```
mere rosii
```