Sesiunea 11

Agenda

- Recapitulare/Intrebari/Diverse
- Rezolvare subiecte BAC 2025 Iunie
- Exercitii tema

Rezolvare subiecte BAC 2025 Iunie

Subiectul 1

1. Rezolvare:
   - Expresia din enunt are valoarea 6 => 2025%2019=6; 6 + 6 = 12
   - a => 2025/2020 + 5 = 1 + 5 = 6
   - b => 2025/2021 + 8 = 0 + 8 = 8
   - c => 2025%2020 + 5 = 5 + 5 = 10
   - d => 2025%2021 + 8 = 4 + 8 = 12
   - Raspuns corect: d
2. Rezolvare:

```
    f(3)=
    n = 3
        i = 1
        i%2 != 0 => calculam f(0) si la intoarcere afisam 1
            f(0)
                n = 0 => nu facem nimic deoarece 1 <= 0 este fals deci for
nu se executa
            afisam 1 (inseamna ca deja am exclus varianta `d`)
        i = 2
        i%2 == 0 => afisam 2 si calculam f(1)
            f(1)
                n = 1 =>
                    i = 1
                    i % 2 != 0 => calculam f(0) si la intoarcere afisam 1
                    f(0) = nu face nimic
                    afisam 1

    pana acum am afisat 121  si putem vedea ca raspunsul corect este `a`
```

   - Raspuns corect: a
3. Rezolvare:
   - a => sintaxa este incorecta. Cand declaram un tablou multidimensional, pentru fiecare dimensiune trebuie sa folosim operatorul de index []. Deci pentru un tablou bidimensional ar trebui sa avem ceva de genul: <nume_tablou>[dim-1][dim-2]
   - b => Respecta atat sintaxa de a declara un tablou bidimensional cat si faptul ca trebuie sa stocam 100 de numere reale:

        - float => inseamna ca o sa stocam doar numere reale
        - m[4][25]=> inseamna ca tabloul nostru este alcatuit din 4 linii, unde fiecare linie are 25 coloane. Deci in total putem stoca 4x25 elemente adica 100.
    - c => sintaxa gresita, asa ceva nu se poate in C/C++
    - d => conform sintaxei avem un sir de numere intregi deci nu satisface enuntul
    - Raspuns corect: b
4. Rezolvare

```
    Stim ca avem primele 4 coduri:
    135024
    135026
    135028
    135042

Conform cerintei, ultimele doua coduri generate ar fi:
    975864
    975862
```

    - Raspuns corect: c
5. Rezolvare:
    - Definitie circuit:Se numește circuit un drum simplu în care extremitatea inițială și finală sunt egale.
    - Definitie drum: Fie G=(V, U) un graf orientat. Se numește drum în graful G o succesiune de noduri, notată D = (x1 , x2 ,..., xk), cu proprietatea că pentru orice 1≤i<k, (xi,xi+1) este arc în G.
    - Deoarece stim ca graful este orientat inseamna ca muchiile au directie (arce) si pentru ca nu exista cicluri (adica nu poti porni dintr-un varf si sa te intorci la el urmand directia arcelor), inseamna ca putem ordona varfurile intr-o ordine topologice: le asezam pe toate in linie astfel incat toate arcele merg doar spre dreapta (dinspre un varf catre un altul)
    - Cum obtinem maximul de arce?
        - pai daca avem n varfuri, le putem aseza asa: v1,v2,v3,..vn
        - Din v1 putem trasa arce catre toate celelalte v2,v3,...vn
        - Din v2 putem trasa arce catre toate celelalte v3,v4,...vn
        - Din v3 putem trasa arce catre toate celelalte v4,v5,...vn
        - samd
        - Astfel fiecare pereche de varfuri (v_i,v_j) cu i < j are exact un arc orientat de la v_i la v_j
        - Acum daca e sa numaram catre arce poate avea fiecare varf: (n-1)+(n-2)+(n-3)_...+1 = (n* (n-1)) / 2 => (10 *9) / 2 = 45
    - Raspuns corect: b

Subiectul 2

1. Rezolvare:

    - a

```
m = 7, n = 17
nr = 0, i =7

repeta
  x = 1
  cat tim 1 * 1 < 7
     x = 1 + 1 = 2
  cat timp 2 * 2 < 7
     x = 2 + 1 = 3
  cat timp 3 * 3 < 7 => false
  daca 3 * 3 = 7 => false => i = 7 + 1 = 8
pana cand 8 > 17 sau nr != 0 => false
repeta
  x = 1
  cat timp 1 * 1 < 8
        x = 1 + 1 = 2
  cat timp 2 * 2 < 8
      x = 2 + 1 = 3
  cat timp 3 * 3 < 8 => false
  daca 3 * 3 == 8 => false => i = 8 + 1 = 9
pana cand 9 > 17 sau nr != 0 => false
repeta
    x = 1
    cat timp 1 * 1 < 9
        x = 1 + 1 = 2
    cat timp 2 * 2 < 9
        x = 2 + 1 = 3
    cat timp 3 * 3 < 9 => false
    daca 3 * 3 == 9 => true => nr = 9
pana cand 9 > 17 sau nr != 0 => true
scrie nr => 9
```

- Programul afiseaza 9
- Programul afiseaza primul patrat perfect in intervalul [m,n]

- b

```
17, 18
```

- c

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

int main() {
    int m, n;
    cin >> m  >> n;
    int nr = 0, i = m;
    do {
```

```
            int x = 1;
            while (x * x < i) {
                x++;
            }
            if (x*x == i) {
                nr = i;
            } else {
                i++;
            }
        } while (i <= n && nr == 0);

        cout << nr;
        return 0;
    }
```

- d

```
    citește m,n
     (numere naturale nenule, m≤n)
     nr<-0;i<-m
    ┌repetă
    │ x<-1
    │┌daca x*x<i atunci
    ││  ┌execută
    ││  │x<-x+1
    ││  └■cât timp x*x<i
    │└■
    │┌dacă x*x=i atunci nr<-i
    ││altfel i<-i+1
    │└■
    └până când i>n sau nr≠0
    scrie nr
```

2. Rezolvare

```
    Conform enuntului rezulta graful din poza de mai jos.
    Se numește graf eulerian un graf care conține un ciclu eulerian.
    Se numește ciclu eulerian un ciclu care conține toate muchiile grafului.
    Se numește ciclu un lanț simplu în care primul vârf este identic cu
ultimul.
    Se numește lanț o succesiune de vârfuri cu proprietatea că oricare două
vârfuri consecutive sunt adiacente.
    Deci din poza si din definitiile de mai sus avem subgraful cu multimea
nodurilor: {1,2,3} si multimea muchiilor:
        {(1,2), (2,3), (3,1)}
```

3. Rezolvare:

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

        // Nota: doar ce este intre linii trebuie sa scrii pe foaia de la
bac
        //-----------------------------------
        struct prajitura {
            int cod;
            float pret;
            int informatii[3];
        }p;
        //-----------------------------------
        int main() {

            // Nota: partea de mai jos nu trebuie sa o scrii la bac. Eu o
fac doar ca sa testez ca totul merge bine.
            p.cod = 20;
            p.pret = 21.34;
            p.informatii[0] = 1;
            p.informatii[1] = 2;
            p.informatii[2] = 3;

            cout << "Prajitura p are codul " << p.cod << "; pretul: " <<
p.pret << "; si urmatoarele informatii: "<<endl;
            cout << "Tipul glazurii: " << p.informatii[0] << endl;
            cout << "Tipul cremei principale: " << p.informatii[1] << endl;
            cout << "Numar de blaturi: " << p.informatii[2] << endl;
            return 0;
        }
```

Subiectul 3

1. Rezolvare:

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

    int ascendent(int n, int x, int y);

    int main() {
        int n = 827, x=9, y=800;
        cout << ascendent(n, x, y);
        return 0;
    }

    int ascendent(int n, int x, int y) {
        int suma = 0;
        for (int i = x; i<= y; i++) {
            int esteAscendent = 1;
            int copieI = i;
            while (copieI > 0) {
```

```
                int ultimaCifraI = copieI % 10;
                if (ultimaCifraI < n % 10) {
                    esteAscendent = 0;
                    break;
                }
                copieI = copieI/10;
            }
            if (esteAscendent) {
                suma += i;
            }
        }
        return suma;
    }
```

2. Rezolvare

```
    #include <iostream>
    #include <cstring>

    using namespace std;

    int main() {

        char text[201];
        char rezultat[201]="";
        cin.getline(text, 201);
        char* cuvant = strtok(text, " ");
        int areCuvinteLungimePara = 0;
        while (cuvant != NULL) {
            int lungimeCuvant = strlen(cuvant);
            if (lungimeCuvant %2 == 0) {
                areCuvinteLungimePara = 1;
                char cuvantInversat[lungimeCuvant+1];
                //Copiem a doua jumatate in variabila ce va tine cuvantul
inversat
                strcpy(cuvantInversat, cuvant+(lungimeCuvant/2));
                //Adaugam prima jumatate la finalul variabilei ce tine
cuvantul inversat
                strncat(cuvantInversat, cuvant, lungimeCuvant/2);
                //Adaugam caracterul de final de sir
                cuvantInversat[lungimeCuvant] = '\0';
                //Adaugam cuvantul inversat la rezultat
                strcat(rezultat, cuvantInversat);
            } else {
                strcat(rezultat, cuvant);
            }
            strcat(rezultat, " ");
            cuvant = strtok(NULL, " ");
        }
        if (!areCuvinteLungimePara) {
            cout << "nu exista";
```

```
        } else {
            cout << rezultat;
        }
        return 0;
    }
```

3. Rezolvare:

   - a.

```
    O sa implementez un algoritm care mai intai va initializa 2
variabile in care vom salva intervalul in care tanarul este disponibil
dupa care citim fisierul linie cu linie.
    Pentru fiecare interval de pe linia curenta, calculam daca se
suprapune cu intervalul tanarului. Facem asta prin mai intai a calcula
ora minima dintre sfarsitul intervalului cand este
disponibil tanarul si sfarsitul intervalului muzeului dupa care
calculam ora maxima dintre inceputul intervalului cand este disponibil
tanarul si inceputul intervalului muzeului.
    Dupa aceea scadem  din ora minima pe cea maxima si daca obtinem un
numar mai mare decat 0 inseamna ca acel muzeu este convenabil.
    Daca muzeul este convenabil, incrementam un contor ce va salva
numarul total al muzeelor convenabil si salvam pozitia muzeului care a
fost convenabil. Aceasta pozitie va fi actualizata
    cu fiecare muzeu convenabil intalnit.
    Algoritmul este eficient din punct de vedere al timpului de
executie deoarece este efectuata o singura parcurgere a fisierului. In
ceea ce priveste memoria  utilizata, algoritmul nostru este
eficient deoarece din totalul de 2*10^5 numere cate pot fi in fisier,
noi in memorie tinem doar 4: anume intervalul studentului si intervalul
muzeului care este citit rand pe rand pentru fiecare             muzeu
```

   - b.

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

    using namespace std;
    int min(int a, int b);
    int max(int a, int b);

    int main() {
        ifstream fin("bac.in");
        int intervalStartTanar, intervalSfarsitTanar;
        fin >> intervalStartTanar >> intervalSfarsitTanar;
        int muzeeConvenabile = 0;
        int pozitieUltimMuzeu = -1;
```

```
            int pozitieCurentaMuzeu = 0;
            int intervalStartMuzeu, intervalSfarsitMuzeu;
            while (fin >> intervalStartMuzeu >> intervalSfarsitMuzeu) {
                pozitieCurentaMuzeu++;
                int oreSuprapuse = min(intervalSfarsitTanar,
    intervalSfarsitMuzeu) - max(intervalStartTanar, intervalStartMuzeu);\
                if (oreSuprapuse > 0) {
                    muzeeConvenabile++;
                    pozitieUltimMuzeu = pozitieCurentaMuzeu;
                }
            }

            cout << muzeeConvenabile << " " << pozitieUltimMuzeu << endl;
            fin.close();
            return 0;
        }

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

        int max(int a, int b) {
            if (a > b) return a;
            return b;
        }
```

Exercitii tema

- Recomandare
  - Repeta functiile principale pentru manipularea sirurilor:
    - strcpy, strncpy
    - strcat, strncat
    - strcmp, strncmp
    - strtok
    - strchr
    - strstr
    - Daca nu intelegi oricare dintre ele, te rog sa imi zici cu putin timp inainte si pregatesc exercitii care sa te ajute sa le intelegi
    - Aici: https://cplusplus.com/reference/cstring/ le gasesti foarte fain explicate dar pot sa explic si eu cu alte exemple, daca ceva nu e clar.
    - In varianta pe care am rezolvat-o azi, ne-au ajutat foarte mult dupa cum ai vazut
  - Extrage definitiie si formulele pentru grafurile orientate/neorientate de aici:
    - https://www.pbinfo.ro/articole/810/grafuri-neorientate
    - https://www.pbinfo.ro/articole/509/grafuri-orientate
  - Analizeaza felul in care se convertesc instructiunile repetitive
    - Am vorbit si azi dar le ai si aici explicate: https://www.pbinfo.ro/articole/23972/limbajul-pseudocod#intlink-8

- Rezolvare varianta model propus 2026 => https://www.pbinfo.ro/resurse/9dc152/examene/2026/E_d_informatica_2026_sp_MI_C_var_model.pdf