Rezolvare subiecte BAC 2021 Iunie

Subiectul I

1.
   - Rezolvare:

```
    a-> Este adevarat doar pentru cazul in care ambele sunt numere
pare. Raspuns corect.
    b-> Este Adevarata si pentru cazuri in care unul dintre ele este
impar (x = 3, y = 2) deci este o optiune invalida
    c-> Invalida deoarece obtinem adevarat si pentru numere impare (x =
5, y = 3)
    d-> Invalida deoarece obtinem adevarat si pentru cazul in care
ambele sunt pare dar si pentru cazul in care ambele sunt impare.
```

   - Raspuns corect: a
2.
   - Rezolvare:

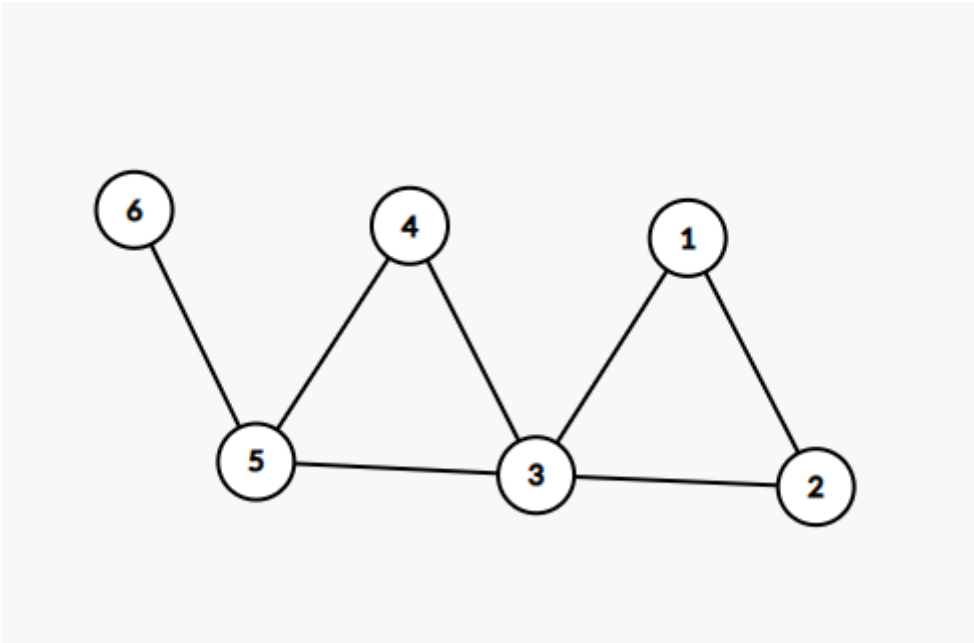
```
    In cazul de fata suntem obligati sa dam valori si sa evaluam
functia.
    a => n = 2021, c = 0
        f(2021, 0) =
            = 1 + 10 * f(202, 0)
                = 2 + 10 * f(20, 0)
                    = f(2, 0)
                        = 2 + 10 * f(0, 0) = 2
                    = 2
                = 2 + 10 * 2 = 22
            = 1 + 10 * 22 = 221
            => Raspuns incorect
    b => n = 200211 si c = 2
        f(200211, 2) =
            = 1 + 10 +f(20021, 2)
                = 1 + 10 * f(2002, 2)
                    = f(200, 2)
                        = 2 + 10 * f(20, 2)
                            = 0 + 10 * f (2, 2)
                                = f(0, 2)
                                    = 0
                                = 0
                            = 0
                        = 2
                    = 2
                = 1 + 10 * 2 = 21
            = 1 + 10 * 21 = 211
            => Raspuns incorect
    c => n =312032, c=3
        f(312032, 3) =
```

```
            = 2 + 10 * f(31203, 3)
                = f(3120, 3)
                    = 0 + 10 * f(312, 3)
                        = 2+ 10 * f(31, 3)
                            = 1+10 * f(3, 3)
                                = f(0, 3)
                                    = 0
                                = 0
                            = 1
                        = 12
                    = 120
                =  120
            = 122
                -> Raspuns incorect
    d => n = 720721, c = 7
        f(720721, 7) =
            = 1 + 10 * f(72072, 7)
                = 2 + 10 * f(7207, 7)
                    = f(720, 7)
                        = 0 + 10 * f(72, 7)
                            = 2 + 10 * f(7, 7)
                                = f(0, 7)
                                    = 0
                                = 0
                            = 2
                        = 20
                    = 20
                = 202
            = 2021
```

- Raspuns corect: `d`

3.
   - Rezolvare:
     - Stim ca un element se afla pe diagonala secundara daca `i+j = n-1`
     - La d avem i = 42 si j = 57 => 42+57 = 99 = 100-1. Rezulta ca d este varianta corecta. De asemenea celalalte sunt invalide si din punct de vedere sintactic.
   - Raspuns corect: `d`
4.
   - Rezolvare:
     - Ciclu elementar: Se numește ciclu un lanț simplu în care primul vârf este identic cu ultimul. Dacă toate vârfurile sunt distincte, mai puțin primul și ultimul, se numește ciclu elementar.

- Conform enuntului avem urmatorul graf



- Prin urmare observam ciclul elementar `1, 2, 3, 1`

- Raspuns corect: `b`

5. - Rezolvare
   - Deoarece stim ca nodurile de pe acelasi nivel au acelasi numar de fii si ca nu exista doua nivele cu acelasi numar de noduri, putem avea urmatoarea distributie:
     - Nivelul 1 => 2 noduri
     - Nivelul 2 => 4 noduri
     - Nivelul 3 => 8 noduri
   - Raspuns corect: `c`

Subiectul II

1. - a

```
x = 8, y = 5
x = 5
y = 8
nr = 1
i = 8
    scrie 1
    1 >=5 false
    nr = 1 * 3 = 3
    scrie 1
i = 7
    scrie 1
    3>= 5 false
    nr = 3 * 3 = 9
    scrie 1
i = 6
    SCRIE 1
    9 >= 5 => scrie 2
    nr = 9 * 3 = 27
    scrie 1
```

```
i = 5
    SCRIe 1
    27 >= 5 => scrie 2
    nr = 27 * 3 =  18
    scrie 1
```

- Programul afiseaza: 1111121121

- b
  - Rezolvare:
    - 15 si 6
- c:

```
#include <iostream>

using namespace std;

int main()
{
    int x, y;
    cin >> x >> y;
    if (x > y) {
        int aux = x;
        x = y;
        y = aux;
    }
    int nr = 1;
    for (int i = y; i>= x; i--) {
        cout << 1;
        if (nr >= x) {
            cout << 2;
        }
        nr = nr * 3;
        cout << 1;
    }
    return 0;
}
```

- d

```
citește x,y
(numere naturale nenule)
┌dacă x>y atunci x<-y
└■
nr<-1
i<- y
┌cat timp i>=x execută
│ scrie 1
│┌dacă nr≥x atunci
│└ scrie 2
```

```
    |└■
    | nr<-nr*3
    | scrie 1
    | i <- i-1
    └■
```

2. ○ Rezolvare:

```
    Pentru usurinta am notat:
    cinteză ciocârlie mierlă privighetoare scatiu
       0        1        2        3          4

    Stim ca 2 si 3 nu trebuie sa fie in acelasi grup

    Primele 4 solutii sunt:
    [cinteză, ciocârlie]
    [cinteză, ciocârlie mierlă],
    [cinteză, ciocârlie, mierlă, scatiu], [cinteză, ciocârlie,
privighetoare]

    Adica
    [0 1],
    [0 1 2],
    [0 1 2 4],
    [0 1 3],

    Ce urmeaza dupa:
    [ciocârlie, privighetoare, scatiu]
    [1, 3, 4]
    |
    [1, 4] -> [ciocârlie, scatiu]
    [2, 4] -> [mierla, scatiu]
```

3. ○ Rezolvare:

```
    if (f.b == 2021){
        fs.a = 2020 - f.a;
        fs.b = 2021;
    } else {
        fs.a = f.a * 2021-f.b*(2020)
        fs.b = f.b * 2021
    }
```

○ Explicatie

```
    In else, deoarece tratam cazul cand numitorii sunt diferiti, am
aplicat principiul inmultirii cu conjugatul. Adica daca avem 2 fractii
cu numitori diferiti: a/b  si c/d, pentru a le aduce la acelasi
numitor, inmultim fiecare fractie cu conjugatul numitorului celeilalte
fractii

    a/b*d/d si c/d*b/b si vom obtine doua fractii cu acelasi numitor bd
```

Subiectul III

1. ◦ Rezolvare:

```
#include <iostream>

using namespace std;

int estePrim(int n);
void divPrim(int n, int& s);

int main()
{
    int s;
    divPrim(16,s);
    cout << s;
    return 0;
}

void divPrim(int n, int& s) {
    int suma = 0;
    for (int i = 2; i<= n; i++) {
        if (n % i == 0 && estePrim(i)) {
            int putere = 0;
            while (n % i == 0){
                putere++;
                n = n / i;
            }

            if (putere % 2 == 1) {
                suma += i;
            }
        }
    }
    s= suma;
}

int estePrim(int n) {
    if (n < 2) {
        return 0;
    }
```

```
        int rezultat = 1;
        for (int i = 2; i*i <= n; i++) {
            if (n % i == 0) {
                rezultat = 0;
                break;
            }
        }
        return rezultat;
    }
```

2. ◦ Rezolvare:

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

    using namespace std;

    int esteVocala(char ch);

    int main()
    {
        int n, k;
        cin >> n >> k;

        char cuvinte[n][11];
        int contor = 0;
        for(int i = 0; i <n;i++) {
            char cuvant[11];
            cin >> cuvant;
            if (esteVocala(cuvant[strlen(cuvant)-1])){
                contor++;
            }
            strcpy(cuvinte[i], cuvant);
        }

        if (contor < k) {
            cout << "nu exista";
        } else {
            for(int i = 0; i < n && k > 0; i++) {
                char ultimulCaracter = cuvinte[i]
[strlen(cuvinte[i])-1];
                if (esteVocala(ultimulCaracter)){
                    cout << cuvinte[i] << endl;
                    k--;
                }
            }
        }
        return 0;
    }


    int esteVocala(char ch) {
```

```
        return strchr("aeiou", ch) != NULL;
    }
```

3.
   - Rezolvare:
     - a:

```
    Mai jos am elaborat un algoritm eficient din punct de vedere
al timpului de executie deoarece efectuam o singura parcurgere a
fisierului si la final vom avea direct rezultatul. In acelasi
timp, algoritmul este eficient din punct de vedere al memoriei
ocupate, deoarece nu folosim vreo structura de date pentru a salva
numere, intrucat din maximul de 10^5 numere, noi in memorie vom
tine 4 variabile ajutatoare si inca una in care vom avea numarul
curent citit.
    Algoritmul va parcurge fisierul si de fiecare data cand va
gasi 2 numere consecutive care indeplinesc conditia, va updata
cele 2 variabile ajutatoare `candidatTermen1` si
`candidatTermen2`. Atunci cand vom gasi un numar care are ca sufix
pe x dar care nu este consecutiv, salvam prima pereche de termeni
valizi.
    La final, in cazul in care nu am gasit minimum 2 termeni
valizi, se va afisa "nu exista", altfel se vor afisa cei 2
termeni.
```

     - b:

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

    using namespace std;
    int esteSufix(int numar, int sufix);



    int main()
    {
        ifstream fin("bac.txt");
        int x;
        fin >> x;
        int termen1 = -1, termen2 = -1;
        int candidatTermen1=-1, candidatTermen2 = -1;
        int ultimaPozitie = 0, pozitieCurenta = 0;
        int numar;
        while (fin >> numar) {
            pozitieCurenta++;
            if (esteSufix(numar, x)) {
                if (candidatTermen1 == -1) {
                    candidatTermen1 = numar;
                    ultimaPozitie = pozitieCurenta;
```

```
                } else if (candidatTermen2 == -1 && pozitieCurenta
== (ultimaPozitie + 1)) {
                    candidatTermen2 = numar;
                    ultimaPozitie=pozitieCurenta;
                } else if (pozitieCurenta == ultimaPozitie + 1) {
                    candidatTermen1 = candidatTermen2;
                    candidatTermen2 = numar;
                } else {
                    // suntem pe un numar care este sufix dar nu e
consecutiv
                    termen1 = candidatTermen1;
                    termen2 = candidatTermen2;
                    candidatTermen1 = numar;
                    ultimaPozitie = pozitieCurenta;
                }
            }
        }

        if (termen1 == -1 || termen2 == -1) {
            cout << "nu exista";
        }else {
            cout << termen1 << " " << termen2;
        }
        fin.close();
        return 0;
    }


    int esteSufix(int numar, int sufix) {
        int rezultat = 1;
        while(sufix > 0) {
            int ultimaCifraNumar = numar % 10;
            int ultimaCifraSufix = sufix % 10;

            numar /= 10;
            sufix /= 10;
            if (ultimaCifraNumar != ultimaCifraSufix || numar ==
0){
                rezultat = 0;
                break;
            }
        }

        return rezultat;
    }
```