Rezolvare subiecte propuse Testul 11 si Testul 10 2021

Testul 11

Subiectul I

1.
   - Rezolvare:
     - a -> Aici nu putem obtine 1 daca 2021 este divizor de al lui X. De exemplu daca luam X = orice multiplu de al lui 2021, sa zicem X = 6063, avem `6063 / (6063/2021) = 2021` deci nu avem cum sa ajungem la 0
     - b -> Pentru a avea rezultatul 0, trebuie ca `x%2021` sa fie mai mare decat x. Si daca 2021 este divizor de al lui x, nu avem cum sa ajungem la cazul asta.
     - c -> Pentru un X care e multiplu de 2021, putem sa ajungem la valoarea 1. De exemplu daca luam x = 6063, obtinem `1`
     - d -> Daca avem x = multiplu de al lui 2021, nu avem cum sa obtinem 1.
   - Raspuns corect: `c`
2.
   - Rezolvare:

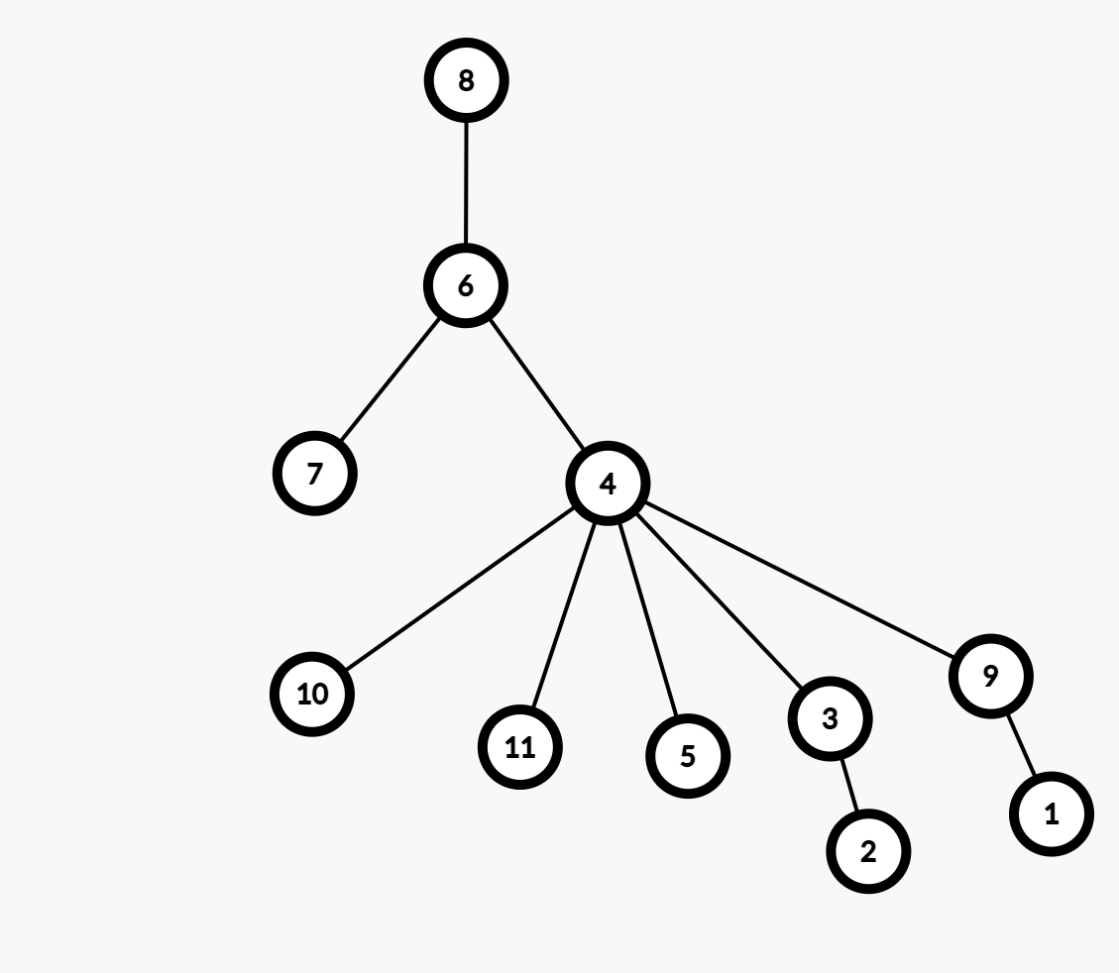
```
0
1
10
100
1000
1001
1002
101
1010
1011
1012
102
```

   - Raspuns corect: `b`
3.
   - Rezolvare:
     - Din punct de vedere sintactic, optiunea valida este a
   - Raspuns corect: `a`
4.
   - Rezolvare:
     - Conform vectorului de tati avem:

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

- Radacina: 8
- 8 parinte pentru: 6
- 6 parinte pentru: 4 si 7
- 4 parinte pentru: 3 5 9 10 11
```

```
    - 3 parinte pentru: 2
    - 9 parinte pentru 1
```

- Rezulta arborele de mai jos:



- Prin urmare observam ca nodul 4 are 5 descendenti directi

- Raspuns corect: c

5.
   - Rezolvare:
     - Stim ca avem un graf neorientat complet cu 210 muchii
     - Teorema: Fie G=(X, U) un graf neorientat. Graful G se numește graf complet dacă oricare două vârfuri distincte ale sale sunt adiacente.
       - Pentru formula: https://www.pbinfo.ro/articole/810/grafuri-neorientate#intlink-7
     - Deci avem ca (n*(n-1))/2 = 210 -> n = 21
   - Raspuns corect: d

Subiectul II

1.
   - a

```
m = 2
n = 9
2 > 9 false

m % 2 == 0 adevarat
    m = m+1 = 3
```

```
cat timp m <= n executa
    m = m+2 = 5
    scrie '*'
cat timp m<= n executa
    m = m+2 = 7
    scrie '*'
cat timp m<= n executa
    m = m+2 = 9
    scrie '*'
cat timp m<= n executa
    m = 11
    scrie '*'
ne oprim
```

- Programul afiseaza: ****
- b: 1 si 158 - Practic trebuie sa urmarim sa fie 40 de pasi, deoarece mergem din 2 in 2.
- c

```
#include <iostream>

using namespace std;

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

    if(m > n) {
        int aux = m;
        m = n;
        n = aux;
    }
    if (m % 2 == 0) {
        m = m+1;
    }
    while (m <=n) {
        m = m+2;
        cout << '*';
    }
    return 0;
}
```

- d

```
citește m,n
(numere naturale)
┌dacă m>n atunci
│ n<->m
└■
```

```
┌dacă m%2=0 atunci
│ m<-m+1
└■
┌ daca m<=n atunci
│ ┌execută
│ │ m<-m+2
│ │scrie ‘*’
│ └■ cat timp m<=n
└■
```

2. ○ Rezolvare:

```
f(2) = 2
f(21) =
    21 - f(19)
        = 19 - f(17) =
            = 17 - f(15)
                = 15 - f(13)
                    = 13 - f(11)
                        = 11 - f(9)
                            = 9 - f(7)
                                = 7 - f(5)
                                    = 5 -f(3)
                                    = 2
                                = 5
                            = 9 - 5 = 4
                        = 11 - 4 = 7
                    = 13 - 7 = 6
                = 15 - 6 = 9
            = 17 - 9 = 8
        = 19 - 8 = 11
    = 21 - 11 = 10
```

3. ○ Rezolvare:

```
    strcpy(x,"bac2021"); // x = "bac2021"
    cout<<x+3<<endl; | printf("%s\n",x+3);c//afisam 2021
    for(i=0;i<strlen(x);i++)
        if(strchr("0123456789",x[i])==0) // pentru fiecare
caracter care nu e cifra afisam caracterul + !
            cout<<x[i]<<'!'; | printf("%c! ",x[i]);
    // practic afisam:
    // 2021
    // b!a!c!
```

Subiectul III

1. ○ Rezolvare:

```cpp
#include <iostream>

using namespace std;

void imog(int x, int y, int& rez);

int main() {
    int x = 40;
    int y = 86;
    int rez;
    imog(x, y, rez);

    cout << rez;
    return 0;
}

void imog(int x, int y, int& rez) {
    int xFaraPare = 0;
    int yFaraPare = 0;
    int px = 1, py = 1;
    while (x) {
        int ultimaCifra = x % 10;
        if (ultimaCifra % 2 != 0) {
            xFaraPare = ultimaCifra * px + xFaraPare;
            px = px * 10;
        }
        x = x/10;
    }

    while (y) {
        int ultimaCifra = y % 10;
        if (ultimaCifra % 2 != 0) {
            yFaraPare = ultimaCifra * py + yFaraPare;
            py = py * 10;
        }
        y = y / 10;
    }

    // Tratam cazul in care numerele nu au nici un numar impar, caz cand rez va fi 0
    if (yFaraPare == 0 || xFaraPare == 0) {
        rez = 0;
        return;
    }

    // calculam oglinditul oricaruia dintre cele 2 numere
    int oglindit = 0;
    while (yFaraPare) {
        int ultimaCifra = yFaraPare % 10;
        oglindit = oglindit * 10 + ultimaCifra;
        yFaraPare = yFaraPare / 10;
    }
```

```
        if (oglindit == xFaraPare) {
            rez = 1;
        } else {
            rez = 0;
        }
    }
```

2. ◦ Rezolvare:

```
    #include <iostream>

using namespace std;

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

    int lungimeSecventa = k - 1; // cate elemente avem de
schimbat de pe fiecare linie sau coloana;
    for(int i =0; i < lungimeSecventa; i++) {
        int aux = matrice[k-1][i];
        matrice[k-1][i] = matrice[i][k-1];
        matrice[i][k-1] = aux;
    }


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

3. ◦ a

```
    Mai jos avem un algoritm eficient din punct de vedere al
timpului de executie si al memoriei deoarece utilizand formula
din enunt, putem sa deducem usor pe ce pozitie se afla ultimul
termen, si bineinteles si penultimul termen pe care il citim,
astfel generand usor, numar cu numar, fara a fi nevoie ca sa le
```

```
generam incepand cu 1, si sa le stocam intr-o structura de date,
dupa care sa le afisam invers.
```

- b

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

    int main() {
    ofstream fout("bac.out");
    int x, y;
    cin >> x >> y;
    fout << y << " " << x << " ";
    // Aflam pozitia lui Y
    int pozitieY = (y-x)/2;
    // Deoarece stim pozitia lui Y, si stim ca mai citim un numar
    // putem deduce usor pozitia urmatorului termen, scazand 2
din pozitia lui Y
    int pozitieTermen = pozitieY-2;
    //Deoarece stim pozitia termenului si cat a fost termenul de
dinaintea lui, deducem usor formula
    int termen = x - 2 * (pozitieY-1);
    while (pozitieTermen >= 0) {
        fout << termen <<" ";
        termen = termen - 2 * pozitieTermen;
        pozitieTermen--;
    }
    fout.close();
    }
```

Testul 10

Subiectul I

1.
   - Rezolvare:
     - a: Intersectia intervalului data cu multimea data, este multimea vida, deci a este invalid
     - b: Aici pentru orice numar din reuniune, obtinem 1
     - c: Deoarece stim ca a este variabila reala, putem sa luam de exemplu numarul 2019 care face parte din intervalul inchis [2002, 2020] si pentru care obtinem 0
     - d: Din nou aici avem multimea vida ca rezultat al intersectiei
   - Raspuns corect: b
2.
   - Rezolvare:

```
    afisa(12345) =
        = afisam "+" calculam afis(12) si la intoarcere trebuie
sa afisam a, adica 12345 si cand iesimn din if sa afisam "+"
           = afisam "+" calculam afisa(0) si la intoarcere
```

```
trebuie sa afisam a adica 12 si cand iesim din if sa afisam "+"
                = afisam "++"
            = afisam "12+"
        = afisam "12345+"
```

- Programul afiseaza: "++++12+12345+"
- Raspuns corect: c

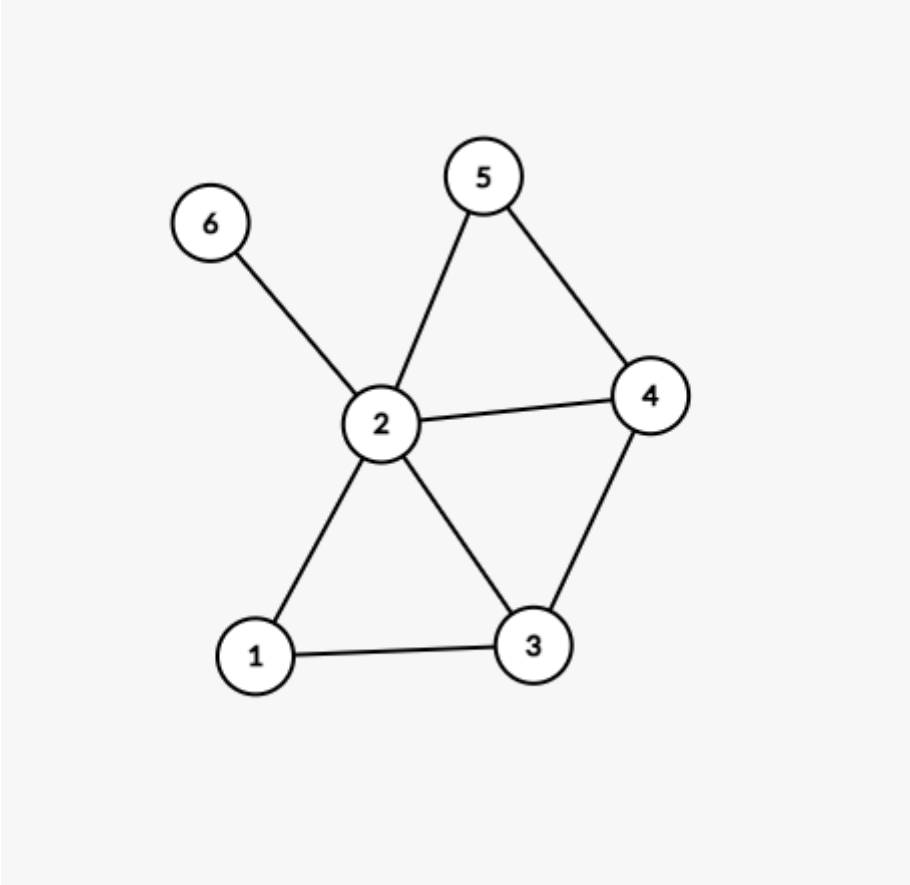
3.
- Rezolvare:

```
Notam:
 0    1    2    3     4      5
Ana Ioana Lia Maria Miruna Simona

Astfel primele 5 solutii generate sunt:
(Ana, Ioana, Lia, Maria)
(Ana, Ioana, Lia, Miruna),
(Ana, Ioana, Lia, Simona),
(Ana, Ioana, Maria, Miruna),
(Ana, Ioana, Maria,Simona)

0 1 2 3
0 1 2 4
0 1 2 5
0 1 3 4
0 1 3 5
```

  - Acum sa evaluam optiunile date:
    - a -> aceasta optiune este invalida deoarece are 5 persoane, si noi trebuie sa avem 4
    - b -> Conform notarii noastre, (Ioana, Maria, Miruna, Simona) este (1,3,4,5) si este valid.
    - c -> Conform notarii noastre, (Lia, Ioana, Maria, Simona) este (2,1,3,5) si aceasta optiune este invalida deoarece cu siguranta inainte de aceasta a fost generata de exemplu 1,2,3,5 si nu avem voie sa refolosim aceiasi membri.
    - d -> Conform notarii noastre, (Maria, Miruna, Lia, Simona) este (3,4,2,5), si la fel ca la c, inaintea de aceasta, cu siguranta am avut (2,3,4,5).
- Raspuns corect: b

4.
- Rezolvare:

- conform enuntului avem graful:



- Din graf vedem ca:
  - Nodul 1: Gradul 2
  - Nodul 2: Gradul 5
  - Nodul 3: Gradul 3
  - Nodul 4: Gradul 3
  - Nodul 5: Gradul 2
  - Nodul 6: Gradul 1

- Raspuns corect: `d`

5.
   - Rezolvare:
     - Din teorie stim ca un arbore are n-1 muchii, si stiind ca avem 4 arbori, rezulta ca avem 16 muchii, deoarece la fiecare arbore, scadem cate o muchie.
   - Raspuns corect: `c`

Subiectul II

1.
   - a

```
x = 2, y = 9
x = -7
y = 2
x = 2 - (-7) = 9
cat timp x>= y
    scrie 'A'
    x = 7
    x % 2 == 0 false => scrie 'B'
cat timp x >= y
```

```
        scrie 'A'
        x = 5
        x % 2 == 0 false => scrie 'B'
    cat timp x>= y
        scrie 'A'
        x = 3
        x % 2 == 0 false => scrie 'B'
    cat timp x >= y
        scrie 'A'
        x = 1
        1 % 2 == 0 false => scrie 'B'
```

- Rezulta ca programul afiseaza: ABABABAB

- b 18 si 19
- c

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

int main() {
    int x,y;
    cin >> x >> y;
    if (x < y) {
        x = x - y;
        y = x + y;
        x = y - x;
    }
    while ( x >= y) {
        cout << 'A';
        x = x - y;
        if (x%2 == 0) {
            cout << 'A';
        } else {
            cout << 'B';
        }
    }
}
```

- d

```
citeşte x,y (numere naturale)
┌dacă x<y atunci
│ x <- x-y; y<-x+y; x<-y-x
└■
┌ daca x≥y atunci
│ ┌execută
│ │ scrie 'A'
│ │ x<-x-y
│ │┌dacă x%2=0 atunci scrie 'A'
```

```
      | ||| altfel scrie 'B'
      | |└■
      | └■ cat timp x≥y
      └■
```

2. ◦ Rezolvare

```
      struct elev{
          int cod;
          float nota1;
          float nota2;
      }y[30];
```

3. ◦ Rezolvare:

```
      strcpy(s,"vorbeste"); // s = vorbeste
      s[3]=s[0]; // -> s =vorveste
      s[5]=s[2]; // -> s =vorverte
      s[0]=s[1]+1; // -> s = porverte
      s[2]=s[1]-2; // -> s = pomverte
      s[6]=s[4]-1; // -> s = pomverde
      strcpy(t,s); t[3]='\0'; // -> t = pomverde, t = pom;
      cout<<t<<endl<<s+3; // afiseaza "pom" si pe linia urmatoare
  "verde"
```

Subiectul III

1. ◦ Rezolvare:

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

      int armonie(int x, int y);


      int main() {
          cout << armonie(8,  12) << endl;
          cout << armonie(8,  13) << endl;
      }

      int armonie(int x, int y) {
          int suma = x+y;
          int sumaDivizoriX = 0;
          for (int i = 1; i<=x; i++) {
              if (x % i == 0) {
                  sumaDivizoriX += i;
              }
```

```
        }
        int sumaDivizoriY = 0;
        for (int i = 1; i<=y; i++) {
            if (y % i == 0) {
                sumaDivizoriY += i;
            }
        }
        if (suma> sumaDivizoriX && suma < sumaDivizoriY) {
            return 1;
        } else {
            return 0;
        }
    }
```

2. ◦ Rezolvare:

```
#include <iostream>
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 frecventa[21] = {0};
    for(int i = 0; i< n-1; i++) {
        int numar = matrice[0][i];
        frecventa[numar]++;
    }

    for (int i = 1; i < m; i++) {
        int numar = matrice[i][n-1];
        frecventa[numar]++;
    }

    for (int i = 2; i< 21; i++){
        if (frecventa[i] == 2) {
            cout << i << " ";
        }
    }
}
```

3. ◦ Rezolvare:
    - a

Mai jos avem un algoritm unde mai intai citim primul termen al sirului, dupa care citim fiecare numar din fisier si in timp ce citim, tinem un contor care va reprezenta cate numere din fisier sunt mai mici sau egale cu primul termen. Dupa care vom afisa acest contor. Programul este eficient din punct de vedere al timpului de executie deoarece fisierul este citit o singura data si in acelasi timp, programul este eficient din punct de vedere al memoriei ocupate deoarece din maximum de 10^5 numere cate pot fi prezente in fisier, noi in memorie tinem doar 2 numere din fisier in orice moment, numere care reprezinta primul termen + numarul curent citit la fiecare pas + o variabila in care vom tine contorul.

- b

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


int main() {
ifstream fin("bac.txt");
int primulTermen;
fin >> primulTermen;
int pozitie = 0;
int numar;
while ( fin >> numar) {
    if (numar <= primulTermen) {
            pozitie++;
    }
}
cout << pozitie;
fin.close();
}
```