Rezolvare varianta 7 din cartea lui Vlad

Subiectul I

1.
   - Rezolvare:
     - Nota: in exercitiile de acest gen, incercam sa eliminam din variante. Conform cerintei, trebuie sa avem valoarea adevarat (1) daca si numai daca x si y reprezinta numere naturale consecutive. Deci daca reusim sa obtinem 1 pentru numere care nu sunt consecutive, inseamna ca trebuie sa eliminam optiunea respectiva. De asemenea, o optiune este eliminata daca nu obtinem 1 pentru numere naturale consecutive
     - Acum sa le luam pe rand:
       - a -> pentru x = 3 si y = 4, obtinem 1 insa obtinem 1 si daca am avea x = 3 si y = 7 deci a pica
       - b -> pentru x = 3 si y = 4, obtinem fals deci pica si optiunea b
       - c -> pentru x= 3 si y = 4, obtinem adevarat. Si pare ca nu obtinem valoarea de adevar pentru alte numere afara de cele consecutive. Momentan e candidata la a fi raspunsul corect.
       - d -> cade din start. Nu obtinem 1 doar pentru numere consecutive. Expresia va fi adevarata si pentru:
         - x = 3, y = 4
         - x = 3, y = 5
         - x = 4, y = 5
         - x = 4, y = 6
         - etc
       - Raspuns corect: c
2.
   - Rezolvare:

```
f(2023, 2022) =
    1 + f(2023, 2023) =
        1 + f(2022, 2023) =
            = 0
        = 1
    = 2
```

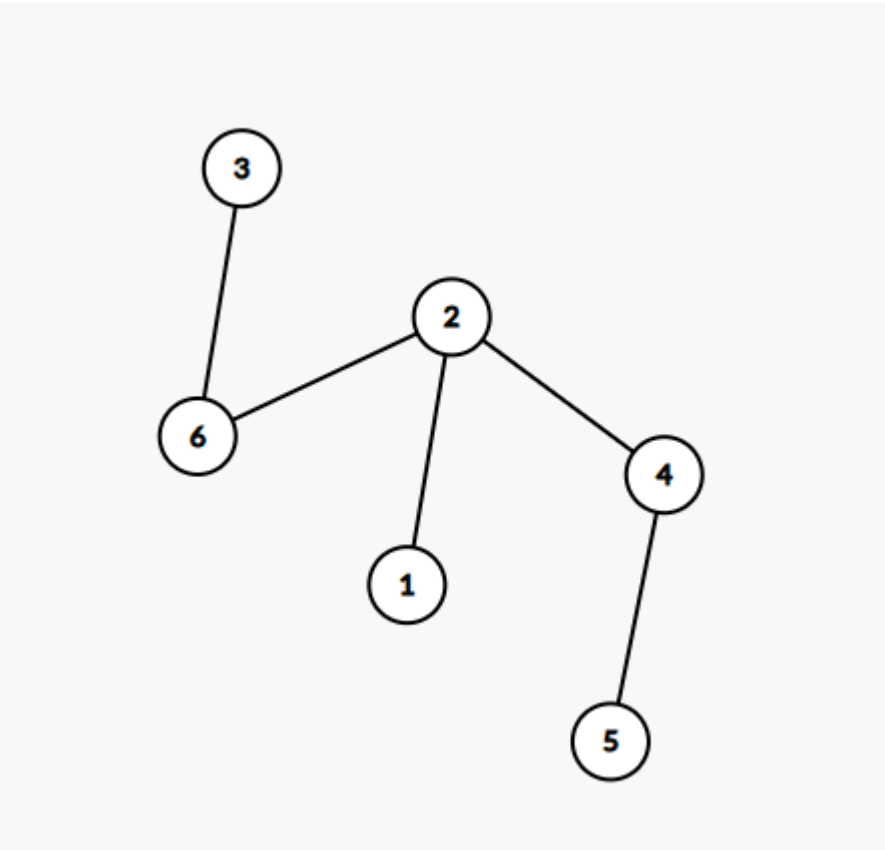
   - Raspuns corect: b
3.
   - Rezolvare:
     - Pentru a fi mai usor de rezolvat, o sa notam cu cifre de la 0 la 6 cele 7 rase de caini astfel:
       - Ciobanesc Anatolian -> 0
       - Ciobanesc Australian -> 1
       - Ciobanesc Belgian -> 2
       - Ciobanesc Carpatin -> 3
       - Ciobanesc Caucazian -> 4
       - Ciobanesc de Bucovina -> 5

        - Ciobanesc German -> `6`
    - Prin urmare, primele 3 solutii date in enunt sunt:
        - `0 1 2 3 4 5 6`
        - `0 1 2 3 4 6 5`
        - `0 1 2 3 5 4 6`
    - Urmatoarele 2 solutii sunt:
        - `0 1 2 3 5 6 4`
        - `0 1 2 4 3 5 6`
    - Nota: logica pare destul de ciudatica, conform raspunsului lor.. ceva de genul (consideram prima cifra, ultima de la dreapta pt a intelege mai usor):
        - la al doilea pas, se muta prima pe a doua pozitie
        - la al treilea pas, se muta prima pe a 3-a pozitie
        - la al patrulea pas, se muta din nou prima pe a doua pozitie
        - la al 5-lea pas, se muta prima pe a 4-a pozitie
    - Raspuns corect: `a`

```
* Ciobanesc Anatolian
* Ciobanesc Australian
* Ciobanesc Belgian
* Ciobanesc Caucazian
* Ciobanesc Carpatin
* Ciobanesc de Bucovina
* Ciobanesc German
```

4.
    - Rezolvare:
        - `strstr` -> va intoarce pozitia de unde se gaseste un sir de caractere in alt sir. De exemplu daca vom face urmatorul apel: `strstr("ana are ac cu ata", "ac")` se va returna `"ac cu ata"`
        - prin urmare, raspunsul este: `vidi,vici` => `b`

5.
   - Conform cerintei, mai jos avem graful ce rezulta din enunt:



   - Rezulta ca nodurile care indeplinesc cerinta sunt:
     - 2
     - 4
     - 6
   - Raspuns corect: b

Subiectul II

1.
   - a

```
x = 52, y = 64
c = 0
repeta
    i = 0
    j = 0
    i+j = 0 -> true
        c = 1
    x = 26*1+0 = 26
    y = 32*1+0 = 32
i*j = 1 fals -> repetam instructiunea
    i = 0
    j = 0
    i+j = 0 -> true
        c = 2
    x = 13*1+0
    y = 16*1+0
i*j = 1 fals -> repetam instructiunea
    i = 1
```

```
        j = 0
        i+j = 0 fals
        x = 6*0+13*1 = 13
        y = 8*1+0 = 8
    i*j = 1 fals -> repetam instructiunea
        i = 1
        j = 0
        i+j = 0 -> fals
        x = 6*0+13*1 = 13
        y = 4*1+0 = 4
    i*j = 1 fals -> repetam instructiunea
        i = 1
        j = 0
        i+j = 0 -> fals
        x = 6 * 0+13*1 = 13
        y = 2*1+0 = 2
    i*j = 1 fals -> repetam instructiunea
        i = 1
        j = 0
        i+j = 0 -> fals
        x = 6 * 0+ 13 * 1 = 13
        y = 1*1+0 = 1
    i * j = 1 fals -> repetam instructiunea
        i = 1
        j = 1
        i+j = 0 dals
        x = 13
        y = 1
    i * j = 1 deci ne oprim
    afisam c -> afisam 2
```

- b
  - Programul afiseaza cea mai mica putere a lui 2, din descompunerea in factori primi a celor doua numere
  - Prin urmare, trebuie sa dam doua valori care atunci cand le descompunem in factori primi, sa nu contina 2
  - 243 si 125 (am "trisat" putin, am luat doar numere care nu au 2 in descompunerea in factori, ca sa fie mai usor)
- c

```cpp
#include <iostream>

using namespace std;

int main() {
    int x, y;
    cin >> x;
    cin >> y;
    int c = 0;
    int i,j;
```

```
        do {
            i = x % 2;
            j = y % 2;
            if (i+j == 0) {
                c = c+1;
            }
            x = x/2 * (1 - i) + x * i;
            y = y /2 * (1 - j) + y * j;
        } while (i*j != 1);
        cout << c;
        return 0;
    }
```

- d

```
    citest x, y ( numere naturale nenule)
    c <- 0
    cat timp (i * j != 1) executa
        i <- x%2
        j <- y%2
        daca i+j = 0 atunci c<- c+1
        x <- [x/2]*(1-i) + x * i
        y <- [y/2] * (1-j) + y * j
    scrie c
```

2. - Rezolvare:
     - nota:
       - observam urmatoarele:
         - elementele de pe diagonala principala sunt toate egale cu 0
         - elementele de pe diagonala secundara sunt egale cu numarul linii + 1
         - restul elementelor sunt egale cu diferenta dintre 4 si numarul coloanei
         - asta daca plecam de la 0.daca plecam de la 1, trebuie o mica adjustare..insa nu e nevoie.
     - solutie:
       - Mai intai solutia pentru oameni normali..care incep de la 0...

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

    using namespace std;

    int main() {
        int matrice[4][4] ={0};
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
            if (i == j) {
```

```
                matrice[i][j] = 0;
            } else if (i+j == 4-1) {
                matrice[i][j] = i+1;
            } else {
                matrice[i][j] = 4 - j;
            }
            }
        }

        // Aici afisam. Este doar pentru noi. Pe foaia de examen
nu trebuie sa o scrii.
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                cout << matrice[i][j]<< " ";
            }
            cout << endl;
        }
        return 0;
    }
```

- si aici solutia pentru cei cu nevoi speciale care incep de la 1..adica exact cum iti cere problema..

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

using namespace std;

int main() {
    int matrice[4][4] ={0};
    for (int i = 1; i <= 4; i++) {
        for (int j = 1; j <= 4; j++) {
        if (i == j) {
            matrice[i-1][j-1] = 0;
        } else if (i+j-2 == 4-1) {
            matrice[i-1][j-1] = i;
        } else {
            matrice[i-1][j-1] = 4 - j + 1;
        }
        }
    }

    // Aici afisam. Este doar pentru noi. Pe foaia de examen
nu trebuie sa o scrii.
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            cout << matrice[i][j]<< " ";
        }
        cout << endl;
    }
```

```
    return 0;
}
```

3. ◦ Rezolvare:

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

using namespace std;

int main() {
    char s[100] = "Ana are MeEe si mioare";
    int i = 0;
    while (i < strlen(s)) {
        if (s[i]>= 65 && s[i]<= 90) {
            strcpy(s+i, s+i+1);
        }
        i++;
    }
    cout << s;
    return 0;
}
```

Subiectul III

1. ◦ Rezolvare:

```
#include <iostream>

using namespace std;

float medie(int, int[]);

int main() {
    int x[] = {4, 3, 4, 3, 4, 3, 10};
    int n = 7;

    cout << medie(n,x);
    return 0;
}

float medie(int n, int x[]) {
    float suma = 0.0;
    int contor = 0;
    for (int i = 0; i < n; i++) {
        if ((i+1) % 2 == 1 && x[i] % 2 == 0) {
            suma+=x[i];
            contor++;
        }
```

```
        }
        if (suma > 0) {
            return suma/contor;
        } else {
            return 0.0;
        }
    }
```

2. ◦ Rezolvare:

```
    #include <iostream>

    using namespace std;

    int sumaCifre(int n);
    int areSumaCifrelorPara(int n);

    int main() {
        // Am hardcodat valorile pentru vector pentru a testa mai
repede.
        // Tu va trebui sa citesti de la tastatura elementele
        int n = 6;
        int v[] = {11, 25, 66, 132, 57, 16};


        for (int i =0; i < n-1; i++) {
            for (int j = i+1; j < n; j++) {
                if (areSumaCifrelorPara(v[i]) &&
areSumaCifrelorPara(v[j])) {
                    // sortam descrescator
                    if (v[i] < v[j]) {
                        int temp = v[i];
                        v[i] = v[j];
                        v[j] = temp;
                    }
                } else if (!areSumaCifrelorPara(v[i]) &&
!areSumaCifrelorPara(v[j])) {
                    if (v[i] > v[j]) {
                        int temp = v[i];
                        v[i] = v[j];
                        v[j] = temp;
                    }
                }
            }
        }

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

```
int areSumaCifrelorPara(int n) {
    int suma = sumaCifre(n);
    if (suma % 2 == 0) {
        return 1;
    } else {
        return 0;
    }
}

int sumaCifre(int n) {
    int sum = 0;
    while (n) {
        sum += n % 10;
        n /= 10;
    }

    return sum;
}
```

3.
   - Rezolvare:
     - a

```
    Algoritmul de mai jos parcurge numar cu numar, din
fisierul dat si cu ajutorul unei functii auxiliare, vedem
daca numarul de divizori este impar, caz in care vom
incrementa un contor.
    Daca, la final, cand am terminat de evaluat numerele din
fisierul de intrare, valoarea contorului va fi tot 0, vom
scrie "NU EXISTA", altfel vom scrie valoarea propriu zisa.
    Algoritmul este eficient din punct de vedere al timpului
de executie deoarece se executa o singura parcurgere a
elementelor din fisier, timp in care si incrementam un
contor. In acelasi timp, algoritmul este eficient din punct
de vedere al memoriei deoarece din toate cele maximum 1
milion de numere cate pot fi in fisier, noi o sa avem in
memorie, maximum 1 element si nu vom declara alte structuri
pentru a manipula datele.
```

     - b

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

int numarDivizori(int n);

using namespace std;
int main() {

    ifstream fin("numere.in");
```

```
    ofstream fout("numere.out");
    int contor = 0;
    int numar;
    while (fin >> numar) {
        if (numarDivizori(numar) % 2 == 1) {
            contor++;
        }
    }

    if (contor == 0) {
        fout << "NU EXISTA";
    } else {
        fout << contor;
    }

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

int numarDivizori(int n) {
    int contor = 0;
    for (int i = 1; i <=n; i++) {
        if (n % i == 0) {
            contor++;
        }
    }
    return contor;
}
```