

4Pb_Atestat_C++_Rezolvate

Bilet 40

Se citesc două triplete de numere naturale (**d1,m1,y1**), respectiv (**d2,m2,y2**). Să se verifice dacă cele două triplete reprezintă două date calendaristice valide, în care **d1,d2** = ziua (day), **m1,m2** = luna (month), **y1,y2** = anul (year). Se consideră an bisect orice an care fie este divizibil cu 400, fie este divizibil cu 4 dar nu și cu 100.

Dacă cele două triplete reprezintă două date calendaristice valide, atunci în ipoteza că prima dată reprezintă ziua de naștere a unei persoane, iar a doua dintre ele reprezintă data curentă, să se determine, în ani împliniți, vârsta persoanei la data curentă.

Se garantează că prima dintre date este anterioară celei de-a doua.

Exemplu

| | |
|------------------------|---------------|
| 29 2 2001 30 5 2013 | data invalida |
| 25 4 1995 20 5 2013 | 18 ani |
| 25 6 1995 10 5 2013 | 17 ani |

VARIANTA 1

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

```
struct Data
{
    int zi;
    int luna;
    int an;
};
```

```
bool checkDate(const Data& data)
{
    bool invalidDate = false;
    switch (data.luna)
    {
        case (1): if (data.zi > 31) invalidDate = true; break;
        case (2):
            if (data.an % 400 == 0 || (data.an % 4 == 0 && data.an % 100 != 0))
            {
                if (data.zi > 29) invalidDate = true;
            }
            else if (data.zi > 28) invalidDate = true;
        case (3): if (data.zi > 31) invalidDate = true; break;
        case (4): if (data.zi > 30) invalidDate = true; break;
        case (5): if (data.zi > 31) invalidDate = true; break;
        case (6): if (data.zi > 30) invalidDate = true; break;
```

```

        case (7): if (data.zi > 31) invalidDate = true; break;
        case (8): if (data.zi > 31) invalidDate = true; break;
        case (9): if (data.zi > 30) invalidDate = true; break;
        case (10): if (data.zi > 31) invalidDate = true; break;
        case (11): if (data.zi > 30) invalidDate = true; break;
        case (12): if (data.zi > 31) invalidDate = true; break;
        default: invalidDate = true;
    }
    return !invalidDate;
}

int main()
{
    std::ifstream f("atestat.txt");
    Data startDate, endDate;

    f >> startDate.zi >> startDate.luna >> startDate.an;
    f >> endDate.zi >> endDate.luna >> endDate.an;

    if (!checkDate(startDate) || !checkDate(endDate))
    {
        cout << "Data invalida";
        return 0;
    }

    else
    {
        int ani = endDate.an - startDate.an;
        if(endDate.luna == startDate.luna)
        {
            if (endDate.zi <= startDate.zi)
                ani--;
        }
        else if(endDate.luna < startDate.luna)
            ani--;

        cout << ani;
    }
}

```

VARIANTA 2

```

#include<fstream>
#include<iostream>
#include <string.h>
using namespace std;
int verif(int d,int m,int y)
{
    if(m==2&& d>28) return 0;
    if(m==4 && d>30) return 0;
    if(m==6&& d>30)return 0;
    if(m==9&& d>30)return 0;
    if(m==11&& d>30)return 0;
}

```