Specialitatea: Programare și testarea produselor program

Disciplina*: Programarea procedurală*  
Lucrare individuală nr. 2

*Executat:* Pricop Maxim  
*Grupa:* P-2413  
*Verificat:* Natalia Gairunova

**Varianta 26**

**Condiție:**

Fișierul text conține următoarele informații despre elev:

* Nume și prenumele fiecărui elev
* Genul fiecărui elev
* Data nașterii fiecărui elev
* 3 note obțiunte în sesiunea de examene de către fiecare elev.

Să se scrie un program care:

1. Citește dintr-un fișier și afișsează informațiile conținute în fișier
2. Numele și prenumele elevului sunt scrise cu majuscule
3. Scrie informațiile despre studenți într-un fișier text nou, în ordinea descrescătoare a punctajului mediu la examene
4. Afișează o listă a elevilor restanțieri, dacă nu sunt astfel de elevi, atunci să fie afișat mesajul corespunzător
5. Determină cuantumul bursei pentru fiecare elev după formula:
   1. 0 lei dacă punctajul medium este sub 7
   2. 150 dacă 7 ≤ media < 8,5
   3. 30 \* medie, dacă media ≥ 8,5

**Rezolvare:**

struct student {

    string nume, prenume;

    char gen;

    string dataNasterii;

    float nota1, nota2, nota3, medie;

    bool restantier = false;

};

// Deschiderea fisierului ce contine informatiile despre elevi

ifstream input("input.txt");

if (!input) {

    cout << "Nu se poate deschide fisierul input.txt" << endl;

    return 0;

}

// Citirea numerelor de elevi din fisier

int nrElevi;

input >> nrElevi;

// Crearea array-ului ce contine elevii din fisier

student elevi[nrElevi];

// Iterarea prin fisier pentru a lua informatiile despre fiecare elev

for (int i = 0; i < nrElevi; i++) {

    // Crearea unei variabile temporare ce va contine datele fiecarui elev pentru o iteratie pentru readability

    student elev; // As fi facut pointer, dar inca nu leam trecut si nu cred ca am voie

    // Citirea informatiilor despre elev de pe o linie

    input >> elev.nume >> elev.prenume >> elev.gen >> elev.dataNasterii >> elev.nota1 >> elev.nota2 >> elev.nota3;

    // Calcularea mediei

    elev.medie = (elev.nota1 + elev.nota2 + elev.nota3) / 3.0;

    // Convertirea numelui si prenumelui in majuscule

    for (int indexLitera = 0; indexLitera < elev.nume.length(); indexLitera++) {

        elev.nume[indexLitera] = toupper(elev.nume[indexLitera]);

    }

    for (int indexLitera = 0; indexLitera < elev.prenume.length(); indexLitera++) {

        elev.prenume[indexLitera] = toupper(elev.prenume[indexLitera]);

    }

    // Scrierea in array a datelor fiecarui elev

    elevi[i] = elev;

}

input.close();

// Afisarea informatiilor citite despre elevi

cout << "Informatii elevi: " << endl;

for (int i = 0; i < nrElevi; i++) {

    cout << elevi[i].nume << " " << elevi[i].prenume << " " << elevi[i].gen << " " << elevi[i].dataNasterii << " "

            << elevi[i].nota1 << " " << elevi[i].nota2 << " " << elevi[i].nota3 << " " << endl;

}

// Sortarea array-ului cu elevii dupa media lor in ordine descrescatoarea cu insertion sort

for (int i = 1; i < nrElevi; i++) {

    student elevCurent = elevi[i];

    int indexInserare = i - 1;

    while (indexInserare >= 0 && elevi[indexInserare].medie < elevCurent.medie) {

        elevi[indexInserare + 1] = elevi[indexInserare];

        indexInserare--;

    }

    elevi[indexInserare + 1] = elevCurent;

}

// Deschiderea/crearea fisierului pentru ouput

ofstream output("output.txt");

if (!output) {

    cout << "Nu se poate deschide/crea fisierul output.txt" << endl;

    return 0;

}

// Scrierea datelor elevilor sortate in fisier

for (int i = 0; i < nrElevi; i++) {

    output << elevi[i].nume << " " << elevi[i].prenume << " " << elevi[i].gen << " " << elevi[i].dataNasterii << " "

            << elevi[i].nota1 << " " << elevi[i].nota2 << " " << elevi[i].nota3 << " " << elevi[i].medie << "\n";

}

output.close();

// Afisarea elevilor restantieri

bool restantieri = false;

cout << endl << "Elevi restantieri:" << endl;

for (int i = 0; i < nrElevi; i++) {

    if (elevi[i].nota1 < 5 || elevi[i].nota2 < 5 || elevi[i].nota3 < 5) {

        cout << elevi[i].nume << " " << elevi[i].prenume << endl;

        elevi[i].restantier = true;

        restantieri = true;

    }

}

if (!restantieri) cout << "Nu sunt elevi restantieri." << endl;

// Calcularea si afisarea burselor pentru fiecare elev

cout << endl << "Elevi cu burse: " << endl;

for (int i = 0; i < nrElevi; i++) {

    float bursa = 0;

    // Calcularea bursei, previne restantierii de la a avea burse

    if (elevi[i].medie < 7 || elevi[i].restantier) continue;

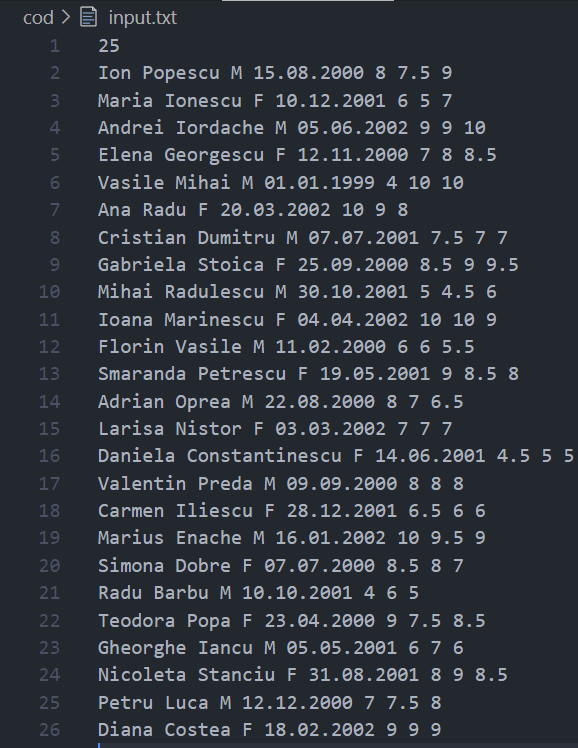
    else if (elevi[i].medie < 8.5) bursa = 150;

    else bursa = 30 \* elevi[i].medie;

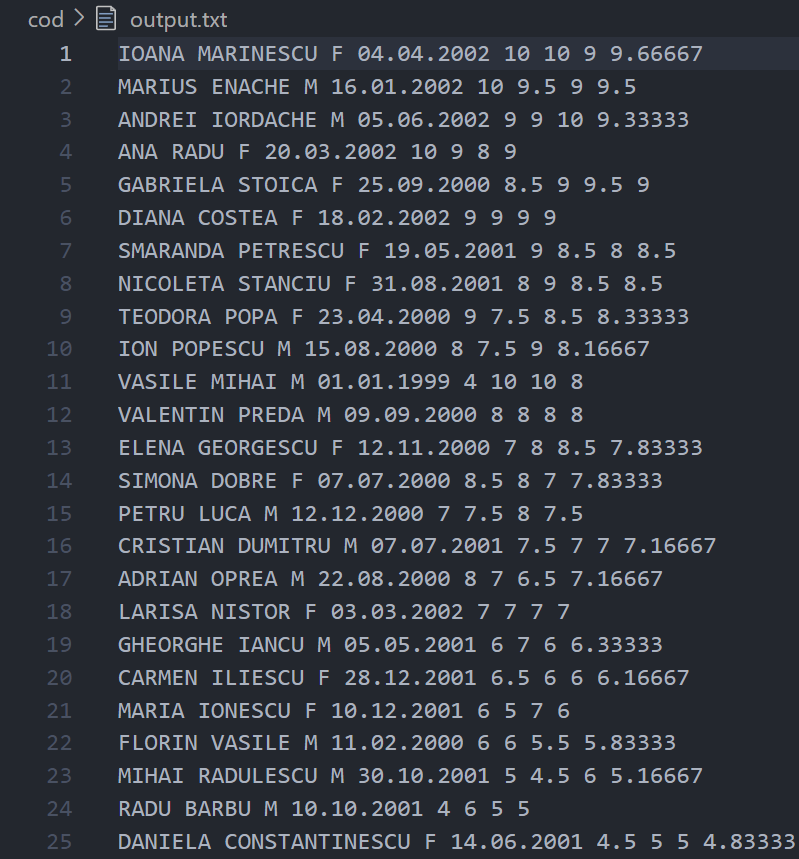
    cout << elevi[i].nume << " " << elevi[i].prenume << " are bursa: " << bursa << endl;

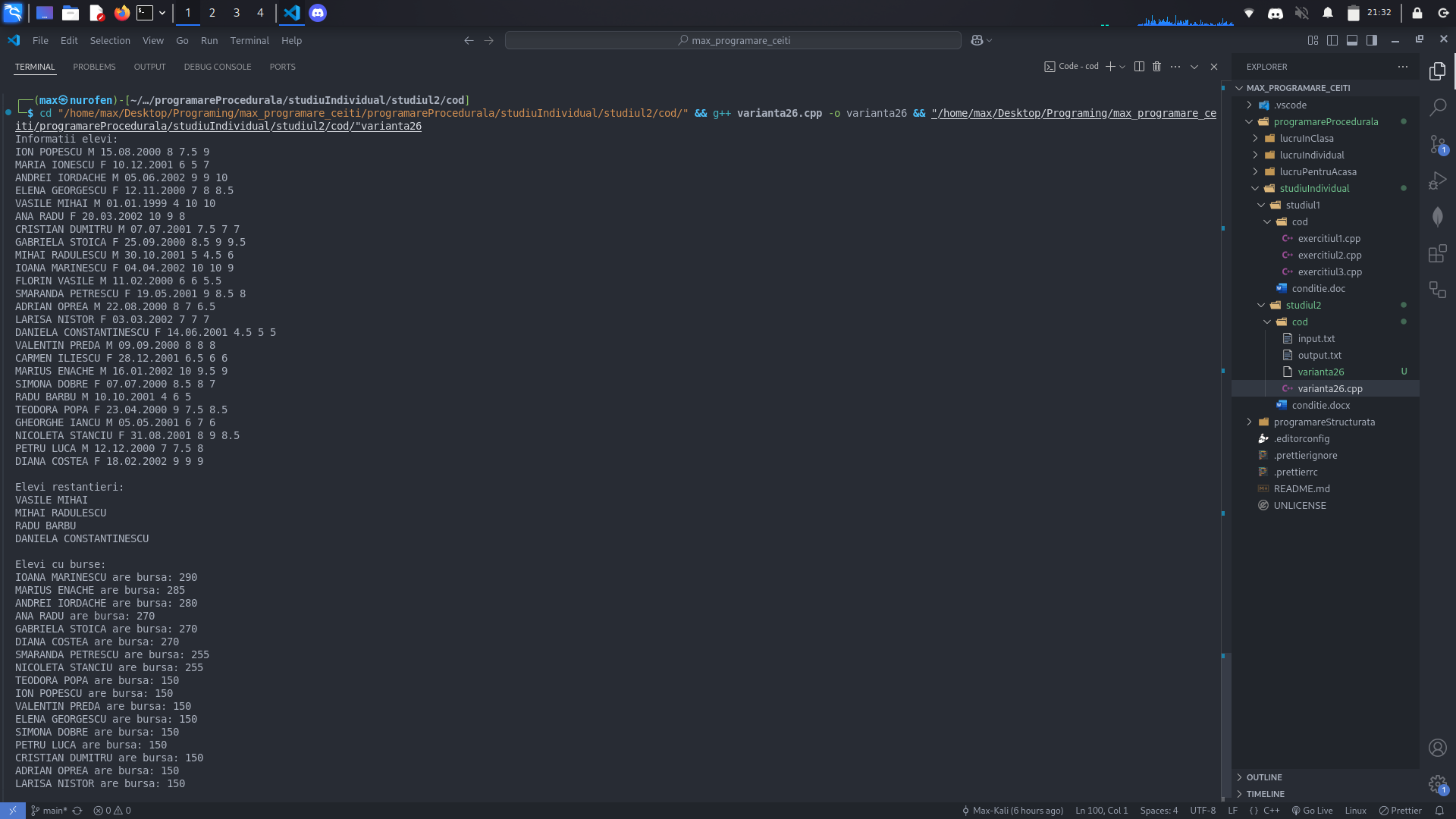
}

**Input.txt:**



**Output.txt:**



**Consolă:**

**Concluzie**

Rezolvarea a arătat că este posibilă dezvoltarea unei aplicații eficiente care citește, prelucrează și scrie date în fișiere, folosind doar array-uri și structuri simple. Transformarea datelor (cum ar fi conversia numelor în majuscule), sortarea în funcție de criterii relevante și aplicarea unor formule condiționate pentru calculul burselor evidențiază flexibilitatea și puterea limbajului C++ atunci când sunt înțelese elementele de bază.

În concluzie, această abordare nu doar că răspunde cerințelor problemei, dar oferă și un exemplu practic despre cum se pot dezvolta soluții software fiabile în domenii din ce în ce mai complexe.

**Literatura și Siteuri Utilizate**

Microsoft Visual Studio Code: https://code.visualstudio.com/

Microsoft Word: <https://www.microsoft.com/en/microsoft-365/word>