Specialitatea: Programare și testarea produselor program

Disciplina*: Programarea structurată*  
Lucrare individuală nr. 2

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

**Partea I – Tablouri unidimensionale**

Eu sunt numărul 26 în catalog, ceea ce înseamnă că trebuie să rezolv problema 26.

**Sarcină:**

Fie dat tabloul H[n], n < 40, de numere întregi.   
Scrieţi un program care determină suma elementelor cu valori pozitive din tabloul H.

Intrare: numărul n şi elementele tabloului H se introduc de la tastatură.  
Ieşire: pe ecran se afişează suma elementelor cu valori pozitive.

**Rezolvare:**

// Declararea variabilei n = numerelor de elemente din array

unsigned n;

// Introducerea numerelor de elemente din array de catre utilizator

cout << "Introdu nr. de elemente ce in array: ";

cin >> n;

// Resetarea numarului sub 40 daca trece de el (din conditie)

if (n >= 40) {

    n = 39;

    cout << "Nr. de elemente trebuie sa fie mai mic decat 40. Valoare introdusa a fost resetata la 39." << endl;

}

// Declararea array-ului si initializarea variabilei ce contine suma numerelor pozitive

int H[n], sumPositive = 0;

// Interarea prin fiecare element al array-ului

for (unsigned i = 0; i < n; i++) {

    // Introducerea la fiecare element individual de catre utilizator

    cout << "Introdu H[" << i << "] = ";

    cin >> H[i];

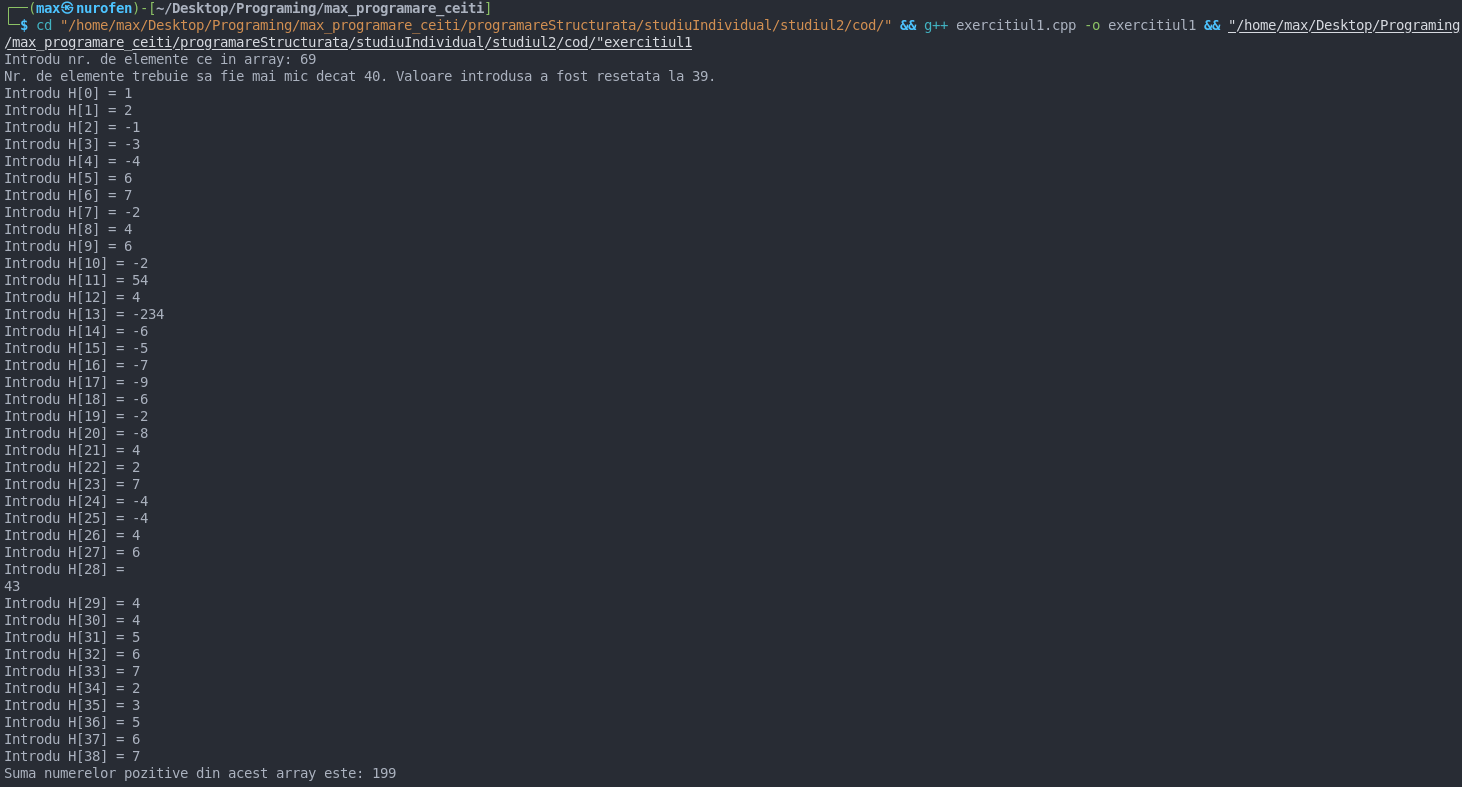
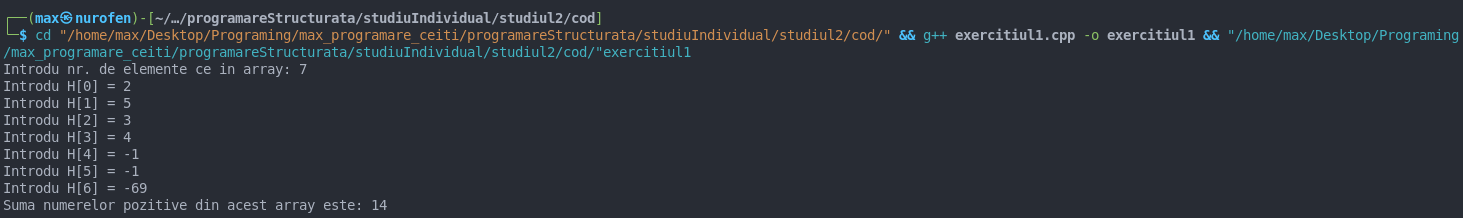
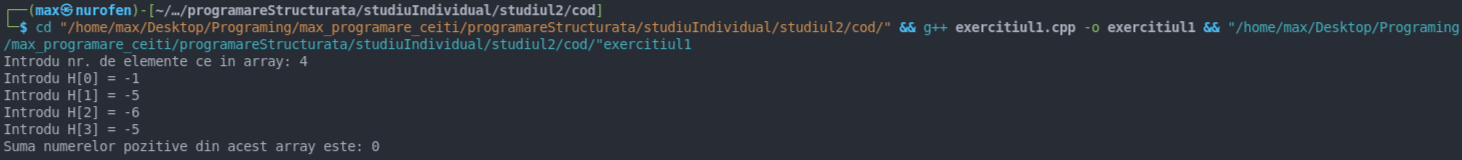
    // Adaugarea elementului daca e pozitiv la suma numerelor pozitive

    if (H[i] > 0) sumPositive += H[i];

};

// Afisarea sumei numerelor pozitive din array

cout << "Suma numerelor pozitive din acest array este: " << sumPositive << endl;

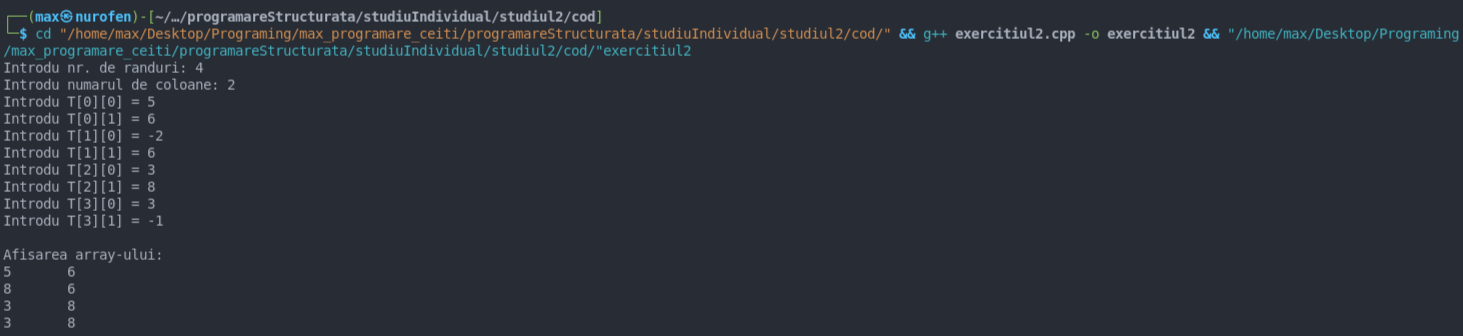
**Rezultat:**

**Partea II – Tablouri bidimensionale**

Eu sunt numărul 26 în catalog, ceea ce înseamnă că trebuie să rezolv problema 26.

**Sarcină:**

Se consideră tabloul T[n][m] cu elemente numere întregi. Să se compună un program care va înlocui în tabloul T elementele lui negative prin elementul maxim. Tabloul modificat se va afişa pe ecran.

**Rezultat:**

**Rezolvare:**

/\*Declararea variabilelor:

n = nr. de randuri (cate elemente contine array-ul principal)

m = nr. de coloane (cate elemente contine fiecare array din array-ul principal)\*/

unsigned n, m;

// Introducerea de randuri si coloane de catre utilizator

cout << "Introdu nr. de randuri: ";

cin >> n;

cout << "Introdu numarul de coloane: ";

cin >> m;

// Declararea array-ului si a variabilei ce va contine numarul maxim din array

int T[n][m], maxNumber;

// Iterarea prin array-ul principal pentru ca utilizatorul sa le acceseze pe cele secundare

for (unsigned row = 0; row < n; row++) {

    // Interarea prin fiecare element al fiecarui array secundar

    for (unsigned column = 0; column < m; column++) {

        // Introducerea fiecarui element din array de catre utilizator

        cout << "Introdu T[" << row << "][" << column << "] = ";

        cin >> T[row][column];

        /\* Determinarea numarului maxim din array:

        Daca este prima iteratie (introducerea primului nr.) nr. maxim este setat ca el

        Daca este alta iteratie, nr. maxim este setat doar daca noul nr. introdus e mai mare ca cel trecut\*/

        if (row == 0 && column == 0) maxNumber = T[0][0];

        else if (T[row][column] > maxNumber) maxNumber = T[row][column];

    };

};

// Text pentru a arata mai frumos

cout << endl << "Afisarea array-ului:" << endl;

// Iterarea prin array-ul principal pentru a afisa elementele la ecran

for (unsigned row = 0; row < n; row++) {

    // Interarea prin fiecare element al fiecarui array secundar

    for (unsigned column = 0; column < m; column++) {

        // Setarea numerelor negative ca si numarul maxim din array (din conditie)

        if (T[row][column] < 0) T[row][column] = maxNumber;

        // Afisarea fiecarui numar la ecran

        cout << T[row][column] << "\t";

    };

    // Rescrie fiecare array iterat pe o linie noua

    cout << endl;

};

**Partea III – Șiruri de caractere**

Eu sunt numărul 26 în catalog, ceea ce înseamnă că trebuie să rezolv problema 26.

**Sarcină:**

Se consideră şirul de caractere S format din litere mici şi mari ale alfabetului latin. Să se compună un program care transformă literele mici în literele respective mari.

**Rezolvare:**

// Declararea variabilei ce va avea lungimea s[];

unsigned lungime;

// Introduerea lungimii de utilizator

cout << "Introdu lungimea sirului s[]: ";

cin >> lungime;

/\*Curatarea bufferului inainte de a citi textul.

Permite utilizatorului sa scrie deoarece normal textul este acoperit de \n

si fara el inputul va fi \n, adica 0 text va fi introdus\*/

cin.ignore();

/\*Declararea array-ului de caractere

E nevoie de lungime + 1 pentru caracterul final null\*/

char s[lungime + 1];

// Introducerea textului

cout << "Introdu sirul de caractere: ";

/\*Limitarea textului la lungimea array-ului + 1 pentru elementul null din array.

Functia get transforma ultimul element valid in null

si permite ca utilizatorul sa introduca cu 1 caracter mai putin decat specificat in ea.\*/

cin.get(s, lungime + 1);

// Iterarea prin fiecare element al s[], cu exceptie ultimul element null de la final

for (unsigned i = 0; i < lungime; i++) {

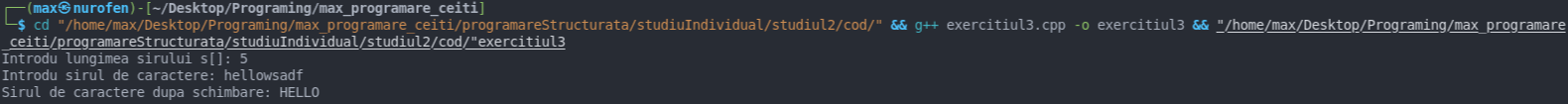
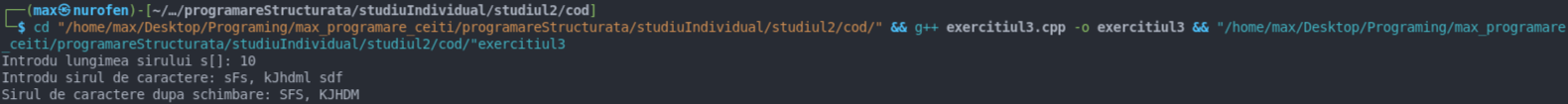
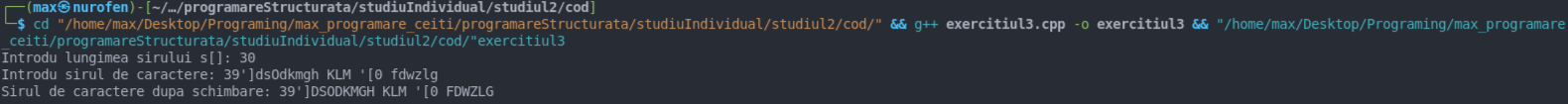
    // Transforma literele mici in litere mari

    if (islower(s[i])) s[i] = toupper(s[i]);

}

// Afisarea rezultatului final

cout << "Sirul de caractere dupa schimbare: " << s << endl;

**Rezultat:**

**Concluzie**

Principala concluzie a lucrării este că utilizarea corectă a array-urilor în C++ permite rezolvarea eficientă a unor probleme variate de procesare a datelor, demonstrând cât de esențiale sunt conceptele fundamentale precum iterarea, validarea și modificarea condiționată. Prin aplicarea acestor principii, se pot dezvolta programe robuste și flexibile, capabile să răspundă cerințelor definite.

Printre principalele puncte discutate se numără realizarea unui program pentru calculul sumei numerelor pozitive dintr-un array unidimensional, unul pentru înlocuirea valorilor negative dintr-un array bidimensional cu cea mai mare valoare prezentă și un al treilea care validează și prelucrează caracterele unui array, transformând literele mici în majuscule. Aceste exerciții au ilustrat aplicabilitatea practică a structurilor de date și a funcțiilor standard pentru prelucrarea corectă și eficientă a informațiilor.

Rezultatele obținute arată că fiecare program a atins obiectivele stabilite, oferind soluții clare și corecte pentru problemele propuse. Mesajul general pe care lucrarea dorește să-l transmită este că stăpânirea tehnicilor de bază în programare, precum gestionarea array-urilor și validarea inputului, constituie fundamentul rezolvării algoritmice a problemelor. Aceste abilități pot fi extinse pentru a aborda sarcini mai complexe, fiind relevante în dezvoltarea de aplicații în diverse domenii.

Impactul lucrării se reflectă în importanța dezvoltării unei gândiri logice structurate, care este esențială nu doar în programare, ci și în alte discipline. Concluziile și rezultatele obținute pot servi drept punct de plecare pentru explorarea unor tehnici mai avansate, contribuind astfel la îmbunătățirea competențelor practice și teoretice.

**Literatura și Siteuri Utilizate**

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

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

Geeks for geeks: https://www.geeksforgeeks.org/cin-get-in-c-with-examples/

Geeks for geeks: https://www.geeksforgeeks.org/how-to-use-the-ignore-function-in-cpp/