

Colegiul National Spiru Haret Tecuci

## OPERATII PE MATRICI STUDIU DE CAZ

Nume: Bors Alin Cristian  
Profesor: Gheorghiu Bogdan  
Clasa: a XII-a C

## Cuprins

<u>1. Descrierea programului</u> .....	3
<u>1.1 Optiunea de sortare</u> .....	3
<u>1.2 Optiunea de calculare a determinantului</u> .....	3
<u>1.3 Operatia de calculare a inversei</u> .....	3
<u>2. Utilizarea programului</u> .....	4
<u>2.1 Calcularea determinantului</u> .....	4
<u>2.2 Calcularea inversei</u> .....	5
<u>2.3 Sortarea unei matrici</u> .....	6
<u>3. Detalii de implementare a programului</u> .....	8
<u>3.1 Compilarea programului</u> .....	8
<u>3.2 Descrierea functionalitatii programului</u> .....	8
<u>4. Bibliografie</u> .....	9

## 1. Descrierea programului

Programul permite utilizatorului sa introduca o matrice de o marime nefixata si ii ofera acestuia posibilitatea de a alege intre cele 3 operatii pe care programul le poate efectua asupra acelei matrici: sortare, calcularea inversei si calcularea determinantului.

Programul nu impune o anumita marime matricii ce va fi introdusa de utilizator. Astfel, se pot efectua operatiile precizate mai sus pe orice tip de matrice, cu precizarea ca operatiile de calculare a determinantului si inversei vor functiona optim doar in cazul matricilor patratice.

### 1.1 Optiunea de sortare

Programul ii ofera utilizatorului posibilitatea de a sorta o matrice fie in functie de elementele de pe o linie, fie in functie de cele de pe o coloana. Acest proces putandu-se realiza atat in ordine crescatoare cat si in ordine descrescatoare.

### 1.2 Optiunea de calculare a determinantului

Utilizatorul poate opta pentru calcularea determinantului matricii introduse. In acest caz programul va afisa valoarea determinantului matricii respective. Doresc sa precizez ca rezultatul va fi unul relevant doar in cazul in care matricea este una patratica, ordinul acesteia nereprezentand o problema.

### 1.3 Operatia de calculare a inversei

Cea de a treia optiunea a programului este aceea de a calcula inversa matricii introduse de utilizator. In cazul in care determinantul matricii respective este 0, atunci programul il va atentiona pe utilizator ca nu exista inversa acelei matrici. Rezultatul afisat de program va fi unul relevant doar in cazul in care matricea este patratica.

## 2. Utilizarea programului

Odata deschis, programul ii va cere utilizatorului sa introduca dimensiunile matricii(fig. 1).

```
*****
size_t minimum;
Bors Alin
Gheorghiu Bogdan
Clasa a XII-a C
*****
introduceti numarul de linii ale matricii: 
```

Fig. 1 – Stadiul de inceput al programului

Dupa ce utilizatorul a introdus marimea matricii, atunci programul ii va cere acestuia sa introduca matricea(fig. 2).

```
introduceti numarul de linii ale matricii: 2
introduceti numarul de coloane ale matricii: 2
introduceti elementul a[0][0]: 1
introduceti elementul a[0][1]: 2
introduceti elementul a[1][0]: 3
introduceti elementul a[1][1]: 4
```

Fig. 2 – introducerea matricii

In acest moment singura informatie necesitata de program este operatia pe care utilizatorul doreste sa o efectueze asupra matricii.

### 2.1 Calcularea determinantului

In continuare dorim sa calculam determinantul matricii introduse, pentru a realiza acest lucru vom scrie cuvantul *determinant* atunci cand programul ne intreaba ce operatie dorim sa realizam asupra matricii introduse de noi(fig. 2).

```

ce operatie doriti sa faceti?
puteti sorta o matrice dupa linii si coloane, calcula inversa unei
matrici si determinantul acesteia: determinant
char comanda[20];

```

Fig. 3 – Optiunea de calculare a determinantului

Dupa aceasta etapa programul ne va afisa valoarea determinantului matricii introduse de noi si ne va intreba daca dorim sa efectuam operatii si pe alta matrice(fig. 4).

```

determinantul este -2.000000
doriti sa efectuati operatii si pe alta matrice?
scrieti da daca da sau nu daca vreti sa inchideti programul: 

```

Fig. 4 – Dialogul de final

## 2.2 Calcularea inversei

Deoarece dorim sa calculam inversa unei matrici de ordinul 4 vom scrie *da*(fig. 5).

```

doriti sa efectuati operatii si pe alta matrice?
scrieti da daca da sau nu daca vreti sa inchideti programul: da
introduceti numarul de linii ale matricii: 4

introduceti numarul de coloane ale matricii: 4

introduceti elementul a[0][0]: 1
introduceti elementul a[0][1]: 2
introduceti elementul a[0][2]: 3
introduceti elementul a[0][3]: 4
introduceti elementul a[1][0]: 5
introduceti elementul a[1][1]: 6
introduceti elementul a[1][2]: 7
introduceti elementul a[1][3]: 8
introduceti elementul a[2][0]: 9
introduceti elementul a[2][1]: 10
introduceti elementul a[2][2]: 11
introduceti elementul a[2][3]: 12
introduceti elementul a[3][0]: 13
introduceti elementul a[3][1]: 14
introduceti elementul a[3][2]: 15
introduceti elementul a[3][3]: 16

```

Fig. 5 – Calcularea inversei unei matrici de 4x4

Pentru ca programul sa calculeze inversa matricii introduse de utilizator vom scrie cuvantul *inversa*.

```

ce operatie doriti sa faceti?
puteti sorta o matrice dupa linii si coloane, calcula inversa unei matrici si dete
rminantul acesteia: inversa

```

Fig. 6 – Optiunea de calcul a inversei

Dupa ce i-am specificat programului ca dorim sa calculam inversa matricii recent introduse, acesta va afisa inversa acesteia(fig. 7).

```
1.000000 2.000000 3.000000 4.000000
5.000000 6.000000 7.000000 8.000000
9.000000 10.000000 11.000000 12.000000
13.000000 14.000000 15.000000 16.000000
```

Fig. 7 – rezultatul dupa calcularea inversei

### 2.3 Sortarea unei matrici

Intr-un final, dorim sa introducem o matrice cu o singura coloana si 4 linii(fig. 8) si sa o sortam dupa coloane. Pentru a realiza acest lucru va trebui sa scriem cuvantul *sortare* atunci cand programul ne va intreba ce operatie dorim sa efectuam asupra matricii(fig. 9).

```
introduceti numarul de linii ale matricii: 4
introduceti numarul de coloane ale matricii: 1
introduceti elementul a[0][0]: 1
introduceti elementul a[1][0]: 2
introduceti elementul a[2][0]: 3
introduceti elementul a[3][0]: 4
```

Fig. 8 – Introducerea unei matrici formata de 4x1

```
ce operatie doriti sa faceti?
puteti sorta o matrice dupa linii si coloane, calcula inversa unei matrici si dete
rminantul acesteia: sortare
```

Fig. 9 – optiunea de sortare a unei matrici

Dupa ce i-am specificat programului ca dorim sa efectuam operatia de sortare acesta ne va intreba daca dori sa sortam elementele matricii dupa linii sau coloane. Deoarece dorim sa le sortam dupa coloane vom scrie cuvantul *coloane*(fig. 10). Daca am fi dorit sa le sortam in functie de linii am fi scris cuvantul *linii*.

```
cum doriti sa sortati matricea? Dupa linii sau dupa coloane?: coloane
cum doriti sa o sortati? crescator sau descrescator: 
```

Fig. 10 – sortam o matrice dupa coloane

Intr-un final, programul ne va cere ordinea in care dorim sa sortam elementele.

Deoarece dorim sa le sortam in ordine crescatoare vom scrie cuvantul *crescator*(fig. 10). Daca am fi dorit sa le sortam in mod descrescator am fi scris cuvantul *descrescator*.

```
cum doriti sa o sortati? crescator sau descrescator: crescator
1.000000 10 Bogdan (n");
2.000000 XII-a Catină);
3.000000 *****\n\n");
4.000000
```

Fig. 10 – Rezultatul sortarii

### 3. Detalii de implementare a programului

Programul este scris in limbajul de programare C si este in conformitate cu standardul **ISO/IEC 9899:1999**. Programul este unul portabil, a fost testat si ruleaza cu succes pe platformele Ubuntu Linux si Windows 7.

Sursa programului a fost compila cu GCC.

#### 3.1 Compilarea programului

Pentru a compila programul vom utiliza compilatorul GCC. Presupunand ca ne aflam in directorul in care este stocata sursa programului, vom executa comanda:

```
gcc -std=c99 -Wall main.c libMatrix/operatii_matrici.c libMatrix/functii_utilitare.c -lm
```

Dupa aceasta etapa, GCC ne va genera fisierul binar si vom putea folosi programul.

#### 3.2 Descrierea functionalitatii programului

In functia *main* a programului ruleaza o bucla *do-while* ce se va opri atunci cand utilizatorul va introduce cuvantul *nu*. La fiecare iteratie a buclei, dupa ce se citesc dimensiunile matricii, se alocă in mod dinamic memorie pentru o matrice de marimea respectiva. Pentru alocarea dinamica a memoriei pentru o matrice se foloseste functia *create\_matrix* care returneaza un pointer spre o structura de tip *Matrix*.

Functia *create\_matrix* este implementata in fisierul *functii\_utilitare.c* din directorul *libMatrix*.

Dupa ce a fost creata o noua matrice programul il va “intreba” pe utilizator ce operatie doreste sa efectueze asupra matricii recent introduse. In functie de raspunsul utilizatorului la aceste intrebari, programul fie va afisa un mesaj de eroare fie va afisa rezultatul operatiei solicitate de utilizator. Mesajul de eroare insemnand ca nu a fost inteleasa operatia dorita de utilizator.

In cazul in care utilizatorul doreste sa calculeze determinantul matricii introduse, se va apela functia *get\_determinant*, implementata in fisierul *operatii\_matrici.c* din directorul *libMatrix*, si se va atribui valoarea returnata de aceasta functie membrului *determinant* al structurii spre care pointeaza matricea.

In cazul in care se doreste calcularea inversei unei matrici, programul va calcula determinantul matricii. Daca acesta este diferit de 0 se va trece la calcularea inversei, in caz contrar se va afisa un mesaj de eroare.



Pentru calcularea inversei se va atribui membrului *inverse* al structurii spre care pointeaza matricea valoarea returnata de functia *compute\_inverse*. Functia *compute\_inverse* returneaza un pointer la o structura de tip *Matrix*. Intr-un final, se va afisa matricea inversa. Pentru a realiza acest lucru se va apela functia *print\_matrix* ce primeste ca si argument un pointer la o structura de tip *Matrix*.

Functia *print\_matrix* este implementata in fisierul *operatii\_matrici.c* din directorul *libMatrix*.

Pentru a putea sorta o matrice au fost implementate 2 functii: una pentru sortarea matricilor dupa linii si alta pentru a le putea sorta dupa coloane. Aceste functii primesc ca si al treilea argument un pointer la functia de comparare. Practic, exista 2 functii de comparare: comparare crescatoare si comparare descrescatoare.

In cazul in care se doreste sortarea unei matrici dupa coloane in ordine crescatoare se va apela functia *sorteazaMatriceaDupaColoane* pasandu-i ca si al treilea argument un pointer la functia *comparaCrescator*. Dupa ce matricea a fost sortata se vor afisa elementele acesteia.

Intr-un final programul va elibera memoria alocata matricii curente folosind functia *distroy\_matrix* si il va intreba pe utilizator daca doreste sa efectueze aceste operatii si pe o alta matrice. In cazul in care raspunsul utilizatorului este *nu* bucla se va opri si programul isi va termina executia. In caz contrar se va incepe o noua iteratie.

## 4. Bibliografie

1. The C Programming Language – Dennis Ritchie si Brian Kernighan
2. Advanced Linux Programming - Mark Mitchell, Jeffrey Oldham,  
si Alex Samuel
3. Introducere in algoritmi – Thomas Cormen
4. Algorimii fundamentali in C++ - Doina Logofatu