**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**Катедра „Компютърни системи“**

**Курсова работа по ПИК1**

**Тема: Матричен калкулатор**

Изготвил: Йордан Беров

Специалност: ИТИ

курс, група: 1-и курс, 36-а група

Фак. №: 501219042

Проверил:

Ас.Ма

**Описание на задачата**

Да се създаде калкулатор за квадратни матрици (макс.размер 4х4), който трябва да извършва следните операции:

1. Събиране, изваждане и умножение на матрици с число или друга матрица.
2. Транспониране на матрица
3. Детерминанта на матрица

**Описание на функциите и начина на изпълнение**

**Main()**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

void processType2(int operation, int additive, int subtractive, int multipliable, int i, int j);

void processType3(int operation, int additive, int subtractive, int multipliable, int i, int j);

void processType4(int operation, int additive, int subtractive, int multipliable, int i, int j);

int main()

{

int operation=1;

int matrixType;

int additive=0, subtractive=0, multipliable=0, i=0, j=0;

float determinant=0;

printf("Please enter the matrix type (1, 2, 3, 4): ");

scanf("%d", &matrixType);

if(matrixType==1)

{

printf("This is not an matrix");

}

else if(matrixType==2)

{

processType2(operation, additive, subtractive, multipliable, i, j);

}

else if(matrixType==3)

{

processType3(operation, additive, subtractive, multipliable, i, j);

}

else if(matrixType==4)

{

processType4(operation, additive, subtractive, multipliable, i, j);

}

else

{

printf("Not a valid order of the matrix");

}

return 0;

}

void processType2(int operation, int additive, int subtractive, int multipliable, int i, int j)//

{

float matrix2[2][2], matrix\_extra2[2][2], multiply\_matrix2[2][2];

float determinant=0;

printf("What would you like to happen with this matrix?: \n");

printf("1.Addition of a matrix and a number\n");

printf("2.Subtraction of a matrix and a number\n");

printf("3.Multiplication of a matrix and a number\n");

printf("4.Addition of a matrix and another matrix\n");

printf("5.Subtraction of a matrix and another matrix\n");

printf("6.Multiplication of a matrix and another matrix\n");

printf("7.Transposition of a matrix and a number\n");

printf("8.Find the determinant of a matrix \n");

printf("0.Close the program\n\n");

printf("Please, enter your chosen operation:");

scanf("%d", &operation);

printf("\n");

printf("Enter the elements of the array:\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("a%d%d:",i,j);

scanf("%f",&matrix2[i][j]);

}

printf("\n");

}

if(operation==1)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &additive);

printf("\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

matrix2[i][j]+=additive;

printf("a%d%d: %0.0f\t", (i+1), (j+1), matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==2)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &subtractive);

printf("\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

matrix2[i][j]-=subtractive;

printf("a%d%d: %0.0f\t \t", (i+1), (j+1), matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==3)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &multipliable);

printf("\n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

matrix2[i][j]\*=multipliable;

printf("a%d%d: %0.0f\t \t", (i+1), (j+1), matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==4)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("b%d%d: ", i,j);

scanf("%f", &matrix\_extra2[i][j]);

printf("\n");

}

}

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

matrix2[i][j]+=matrix\_extra2[i][j];

printf("a%d%d: %0.0f\t \t", (i+1), (j+1), matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==5)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("b%d%d: ", i,j);

scanf("%f", &matrix\_extra2[i][j]);

printf("\n");

}

}

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

matrix2[i][j]-=matrix\_extra2[i][j];

printf("a%d%d: %0.0f\t \t", (i+1), (j+1), matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==6)//Ready//

{

printf("Enter the extra array you wish to use: \n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("B%d%d: ", (i+1),(j+1));

scanf("%f", &matrix\_extra2[i][j]);

printf("\n");

}

printf("\n");

}

int k=0;

for (i = 0; i < 2; i++)

{

for (j = 0; j < 2; j++)

{

multiply\_matrix2[i][j] = 0;

for (k = 0; k < 2; k++)

{

multiply\_matrix2[i][j] += matrix2[i][k]\*matrix\_extra2[k][j];

}

printf("C%d%d: %0.0f ",(i+1), (j+1), multiply\_matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==7)//Ready//

{

printf("Result: \n");

for (j = 0; j < 2; ++j)

{

for (i = 0; i < 2; ++i)

{

printf(" %0.0f", matrix2[i][j]);

}

printf("\n");

}

}

else if(operation==8)//Ready//

{

determinant=(matrix2[0][0]\*matrix2[1][1])-(matrix2[0][1]\*matrix2[1][0]);

printf("The determinant is: %0.2f", determinant);

}

else

{

printf("Error: Command not found.");

}

}

void processType3(int operation, int additive, int subtractive, int multipliable, int i, int j)//

{

float matrix3[3][3], matrix\_extra3[3][3], multiply\_matrix3[3][3];

float determinant=0;

printf("What would you like to happen with this matrix?: \n");

printf("1.Addition of a matrix and a number\n");

printf("2.Subtraction of a matrix and a number\n");

printf("3.Multiplication of a matrix and a number\n");

printf("4.Addition of a matrix and another matrix\n");

printf("5.Subtraction of a matrix and another matrix\n");

printf("6.Multiplication of a matrix and another matrix\n");

printf("7.Transposition of a matrix and a number\n");

printf("8.Find the determinant of a matrix \n");

printf("0.Close the program\n\n");

printf("Please, enter your chosen operation:");

scanf("%d", &operation);

printf("\n");

printf("Enter the elements of the array:\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

printf("A%d%d: ", i+1,j+1);

scanf("%f/t", &matrix3[i][j]);

}

printf("\n");

}

if(operation==1)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &additive);

printf("\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

matrix3[i][j]+=additive;

printf("Result: %0.0f \t", matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==2)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &subtractive);

printf("\n");

printf("Result: \n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

matrix3[i][j]-=subtractive;

printf("%0.0f \t", matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==3)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &multipliable);

printf("\n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

matrix3[i][j]\*=multipliable;

printf("Result: %0.0f \t", matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==4)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<3;i++)

{

for(int j=0;j<3;j++)

{

printf("a%d%d", i,j);

scanf("%f", &matrix\_extra3[i][j]);

}

printf("\n");

}

for(i=0;i<3;i++)

{

for(int j=0;j<3;j++)

{

matrix3[i][j]+=matrix\_extra3[i][j];

printf("Result: %0.0f \t", matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==5)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

printf("a%d%d", i,j);

scanf("%f", &matrix\_extra3[i][j]);

}

printf("\n");

}

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

matrix3[i][j]-=matrix\_extra3[i][j];

printf("Result: %0.0f \t", matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==6)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

printf("a%d%d", i,j);

scanf("%f", &matrix\_extra3[i][j]);

}

printf("\n");

}

int k=0;

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

multiply\_matrix3[i][j] = 0;

for (k = 0; k < 3; k++)

{

multiply\_matrix3[i][j] += matrix3[i][k]\*matrix\_extra3[k][j];

}

printf("C%d%d: %.0f ",(i+1), (j+1), multiply\_matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==7)//Ready//

{

printf("Result: \n");

for (j = 0; j < 3; ++j)

{

for (i = 0; i < 3; ++i)

{

printf(" %0.0f", matrix3[i][j]);

}

printf("\n");

}

}

else if(operation==8)//Ready//

{

for(i=0;i<3;i++)

{

determinant = ((matrix3[0][0]\*((matrix3[1][1]\*matrix3[2][2])-(matrix3[2][1]\*matrix3[1][2]))+(matrix3[0][1]\*((matrix3[1][2]\*matrix3[2][0])-(matrix3[2][2]\*matrix3[1][1])))+

(matrix3[0][2]\*((matrix3[1][1]\*matrix3[2][1])-(matrix3[2][0]\*matrix3[1][1])))));

}

printf("The determinant is: %.0f", determinant);

}

else

{

printf("Error: Command not found.");

}

}

void processType4(int operation, int additive, int subtractive, int multipliable, int i, int j)//

{

float matrix4[4][4], matrix\_extra4[4][4], multiply\_matrix4[4][4];

float determinant=0;

printf("What would you like to happen with this matrix?: \n");

printf("1.Addition of a matrix and a number\n");

printf("2.Subtraction of a matrix and a number\n");

printf("3.Multiplication of a matrix and a number\n");

printf("4.Addition of a matrix and another matrix\n");

printf("5.Subtraction of a matrix and another matrix\n");

printf("6.Multiplication of a matrix and another matrix\n");

printf("7.Transposition of a matrix and a number\n");

printf("8.Find the determinant of a matrix \n");

printf("0.Close the program\n\n");

printf("Please, enter your chosen operation:");

scanf("%d", &operation);

printf("\n");

printf("Enter the elements of the array:\n");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

printf("A%d%d: ", i,j);

scanf("%f", &matrix4[i][j]);

}

printf("\n");

}

if(operation==1)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &additive);

printf("\n");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

matrix4[i][j]+=additive;

printf("Result: %0.0f \t", matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==2)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%d", &subtractive);

printf("\n");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

matrix4[i][j]-=subtractive;

printf("Result: %0.0f \t", matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==3)//Ready//

{

printf("Enter the number you wish to use: ");

scanf("%f", &multipliable);

printf("\n");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

matrix4[i][j]\*=multipliable;

printf("Result: %d \t", matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==4)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

scanf("%f", &matrix\_extra4[i][j]);

}

printf("\n");

}

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

matrix4[i][j]+=matrix\_extra4[i][j];

printf("Result: %0.0f \t", matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==5)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

scanf("%f", &matrix\_extra4[i][j]);

}

printf("\n");

}

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

matrix4[i][j]-=matrix\_extra4[i][j];

printf("Result: %0.0f \t", matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==6)//Ready//

{

printf("Enter the extra array you wish to use: ");

for(i=0;i<4;i++)

{

for(j=0;j<4;j++)

{

scanf("%f", &matrix\_extra4[i][j]);

}

printf("\n");

}

int k=0;

for (i = 0; i < 4; i++)

{

for (j = 0; j < 4; j++)

{

multiply\_matrix4[i][j] = 0;

for (k = 0; k < 4; k++)

{

multiply\_matrix4[i][j] += matrix4[i][k]\*matrix\_extra4[k][j];

}

printf("C%d%d: %.0f ",(i+1), (j+1), multiply\_matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==7)//Ready//

{

for (j = 0; j < 4; j++)

{

for (i = 0; i < 4; i++)

{

printf("%.0f", matrix4[i][j]);

}

printf("\n");

}

}

else if(operation==8)//Ready//

{

determinant=(matrix4[0][0] \* ((matrix4[1][1]\*((matrix4[2][2]\*matrix4[3][3])-(matrix4[3][2]\*matrix4[2][3]))+(matrix4[1][2]\*((matrix4[2][3]\*matrix4[3][1])-(matrix4[3][3]\*matrix4[2][2])))+

(matrix4[1][3]\*((matrix4[2][2]\*matrix4[3][2])-(matrix4[3][1]\*matrix4[2][2]))))));//Minor1

determinant = determinant -(matrix4[1][2] \* ((matrix4[1][0]\*((matrix4[2][2]\*matrix4[3][3])-(matrix4[3][2]\*matrix4[2][3]))+(matrix4[1][2]\*((matrix4[2][3]\*matrix4[3][1])-(matrix4[3][3]\*matrix4[2][2])))+

(matrix4[1][3]\*((matrix4[2][2]\*matrix4[3][2])-(matrix4[3][1]\*matrix4[2][2]))))));//Minor2

determinant = determinant +(matrix4[1][3] \* ((matrix4[1][0]\*((matrix4[2][1]\*matrix4[3][3])-(matrix4[3][1]\*matrix4[2][3]))+(matrix4[1][1]\*((matrix4[2][3]\*matrix4[3][0])-(matrix4[3][3]\*matrix4[2][1])))+

(matrix4[1][3]\*((matrix4[2][1]\*matrix4[3][1])-(matrix4[3][0]\*matrix4[2][1]))))));//Minor3

determinant = determinant -(matrix4[1][4] \* ((matrix4[2][1]\*((matrix4[3][2]\*matrix4[4][3])-(matrix4[4][2]\*matrix4[3][3]))+(matrix4[2][2]\*((matrix4[3][3]\*matrix4[4][1])-(matrix4[4][3]\*matrix4[3][2])))+

(matrix4[2][3]\*((matrix4[3][2]\*matrix4[4][2])-(matrix4[4][1]\*matrix4[3][2]))))));//Minor4

printf("%.0f", determinant);

}

else{

printf("Error: Command not found.");

}

}

**Описание на функцията: processType2();**

Функцията е от тип void и отговаря за изпълнението на всички аритметични дейности на калкулатора за матрици от втори ред. Функцията постига това в няколко стъпки.

1. Програмата пита потребителя, коя операция да изпълни, чрез принтиране на възможните операции с числов код пред всяка от тях. Потребителят въвежда кода на избраната от него операция
2. Чрез два вместени for цикъла потребителят въвежда стойността на всеки член на матрицата.
3. Програмата прави проверка с if и else if за кода, въведен от потребителя.
4. Програмата събира/изважда/умножава матрицата с число, което потребителят е въвел, като с два nested for цикъла върти всеки елемент на матрицата и извършва зададеното му действие с числото.
5. Програмата събира/изважда матрицата с друга матрица, която потребителят е въвел, като с два nested for цикъла върти всеки елемент на матрицата и извършва зададеното му действие със същия член на другата матрица.
6. Програмата умножава матрицата с друга матрица, която потребителят е въвел, като с два nested for цикъла въвежда втората матрица. Програмата задава стойности за всяка стойност на multiply\_matrix2[2][2] = 0. След това всеки член на multiply\_matrix2[2][2] е равен на сбора на multiply\_matrix2[2][2] със произведението на съответните членове на първата и втората матрица.
7. Програмата транспонира матрицата, чрез цикъла

for (j = 0; j < 2; ++j)

{

for (i = 0; i < 2; ++i)

{

printf(" %d", matrix2[i][j]);

}

printf("\n");

}

1. Програмата намира детерминантата на матрицата по формулата (а11\*а22)-(а12\*а21) и печата получения резултат.

**Описание на функцията: processType3();**

Функцията е от тип void и отговаря за изпълнението на всички аритметични дейности на калкулатора за матрици от трети ред. Функцията постига това в няколко стъпки.

1. Програмата пита потребителя, коя операция да изпълни, чрез принтиране на възможните операции с числов код пред всяка от тях. Потребителят въвежда кода на избраната от него операция
2. Чрез два вместени for цикъла потребителят въвежда стойността на всеки член на матрицата.
3. Програмата прави проверка с if и else if за кода, въведен от потребителя.
4. Програмата събира/изважда/умножава матрицата с число, което потребителят е въвел, като с два nested for цикъла върти всеки елемент на матрицата и извършва зададеното му действие с числото.
5. Програмата събира/изважда матрицата с друга матрица, която потребителят е въвел, като с два nested for цикъла върти всеки елемент на матрицата и извършва зададеното му действие със същия член на другата матрица.
6. Програмата умножава матрицата с друга матрица, която потребителят е въвел, като с два nested for цикъла въвежда втората матрица. Програмата задава стойности за всяка стойност на multiply\_matrix3[3][3] = 0. След това всеки член на multiply\_matrix3[3][3] е равен на сбора на multiply\_matrix3[3][3] със произведението на съответните членове на първата и втората матрица.
7. Програмата транспонира матрицата, чрез цикъла

for (j = 0; j < 3; ++j)

{

for (i = 0; i < 3; ++i)

{

printf(" %d", matrix3[i][j]);

}

printf("\n");

}

1. Програмата намира детерминантата на матрицата по формулата на Сарус за матрица 3х3 и печата получения резултат.

**Описание на функцията: processType4();**

Функцията е от тип void и отговаря за изпълнението на всички аритметични дейности на калкулатора за матрици от четвърти ред. Функцията постига това в няколко стъпки.

1. Програмата пита потребителя, коя операция да изпълни, чрез принтиране на възможните операции с числов код пред всяка от тях. Потребителят въвежда кода на избраната от него операция
2. Чрез два вместени for цикъла потребителят въвежда стойността на всеки член на матрицата.
3. Програмата прави проверка с if и else if за кода, въведен от потребителя.
4. Програмата събира/изважда/умножава матрицата с число, което потребителят е въвел, като с два nested for цикъла върти всеки елемент на матрицата и извършва зададеното му действие с числото.
5. Програмата събира/изважда матрицата с друга матрица, която потребителят е въвел, като с два nested for цикъла върти всеки елемент на матрицата и извършва зададеното му действие със същия член на другата матрица.
6. Програмата умножава матрицата с друга матрица, която потребителят е въвел, като с два nested for цикъла въвежда втората матрица. Програмата задава стойности за всяка стойност на multiply\_matrix4[4][4] = 0. След това всеки член на multiply\_matrix4[4][4] е равен на сбора на multiply\_matrix4[4][4] със произведението на съответните членове на първата и втората матрица.
7. Програмата транспонира матрицата, чрез цикъла

for (j = 0; j < 4; ++j)

{

for (i = 0; i < 4; ++i)

{

printf(" %d", matrix4[i][j]);

}

printf("\n");

}

1. Програмата намира детерминантата по метода на минорите. Умножава всеки член на първия ред на матрицата с образувания от него минор. После, програмата вади второ и четвърто произведение и събира първо и трето произведение, така намирайки детерминантата.

Начало на програмата

int matrixType;

int operation=1, additive=0;

int subtractive=0, multipliable=0, i=0, j=0;

float determinant=0;

printf("Please enter the matrix type (1, 2, 3, 4): ");

scanf("%d", &matrixType);

printf("This is not an matrix");

Край

return 0;

Други стойности

processType4(operation, additive, subtractive, multipliable, i, j);

processType3(operation, additive, subtractive, multipliable, i, j);

1

2

printf("This is not an matrix");

4

3

printf("Not a valid order of the matrix");

processType2(operation, additive, subtractive, multipliable, i, j);

MatrixType