Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**Алгоритмізація та програмування 1: Базові концепції програмування**

ЗВІТ

Модульна контрольна робота№1

Варіант № \_7\_\_

Виконала:

Студент групи ТР-12

Каркушевський В.Л

Перевірила :Смаковська Г.М

Дата здачі \_\_\_01.12.21 o 12 :\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2021

**Модульна контрольна робота №1**

***Код програми:***

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include<stdlib.h>

#include <math.h>

int main(int argc, char \*argv[]) {

double Matrix [3][3];

if (argc == 10)

{

int k = 0;

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

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

k++;

Matrix[i][j] = atof(argv[k]);

}

}

}

//вивід почтакової матриці

printf ("Початкова матриця з консолі:\n");

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

printf ("|");

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

printf(" %.2f ", Matrix [i][j]);

}

printf ("|");

printf("\n");

}

printf("\n");

float leftdiagonal = Matrix[0][0]\*Matrix[1][1]\*Matrix[2][2] + Matrix[0][1]\*Matrix[1][2]\*Matrix[2][0] + Matrix[0][2]\*Matrix[1][0]\*Matrix[2][1];

float rightdiagonal = Matrix[0][2]\*Matrix[1][1]\*Matrix[2][0] + Matrix[0][1]\*Matrix[1][0]\*Matrix[2][2] + Matrix[0][0]\*Matrix[1][2]\*Matrix[2][1];

float determinant = leftdiagonal-rightdiagonal;

printf("Детермінант =%.2f \n\n",determinant);

float A00 = Matrix[1][1]\*Matrix[2][2]-Matrix[1][2]\*Matrix[2][1];

float A01 = -1\*(Matrix[1][0]\*Matrix[2][2]-Matrix[1][2]\*Matrix[2][0]);

float A02 = Matrix[1][0]\*Matrix[2][1]-Matrix[1][1]\*Matrix[2][0];

float A10 = -1\*(Matrix[0][1]\*Matrix[2][2]-Matrix[0][2]\*Matrix[2][1]);

float A11 = Matrix[0][0]\*Matrix[2][2] - Matrix[0][2]\*Matrix[2][0];

float A12 = -1\*(Matrix[0][0]\*Matrix[2][1] - Matrix[0][1]\*Matrix[2][0]);

float A20 = Matrix[0][1]\*Matrix[1][2] - Matrix[0][2]\*Matrix[1][1];

float A21 = -1\*(Matrix[0][0]\*Matrix[1][2] - Matrix[0][2]\*Matrix[1][0]);

float A22 = Matrix[0][0]\*Matrix[1][1] - Matrix[0][1]\*Matrix[1][0];

float OBR[3][3];

OBR[0][0] = A00\*1./determinant;

OBR[0][1] = A10\*1./determinant;

OBR[0][2] = A20\*1./determinant;

OBR[1][0] = A01\*1./determinant;

OBR[1][1] = A11\*1./determinant;

OBR[1][2] = A21\*1./determinant;

OBR[2][0] = A02\*1./determinant;

OBR[2][1] = A12\*1./determinant;

OBR[2][2] = A22\*1./determinant;

//вивід оберненої матриці

printf ("Обернена матриця з консолі:\n");

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

printf ("|");

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

printf("%.2f ", OBR[i][j]);

}

printf ("|");

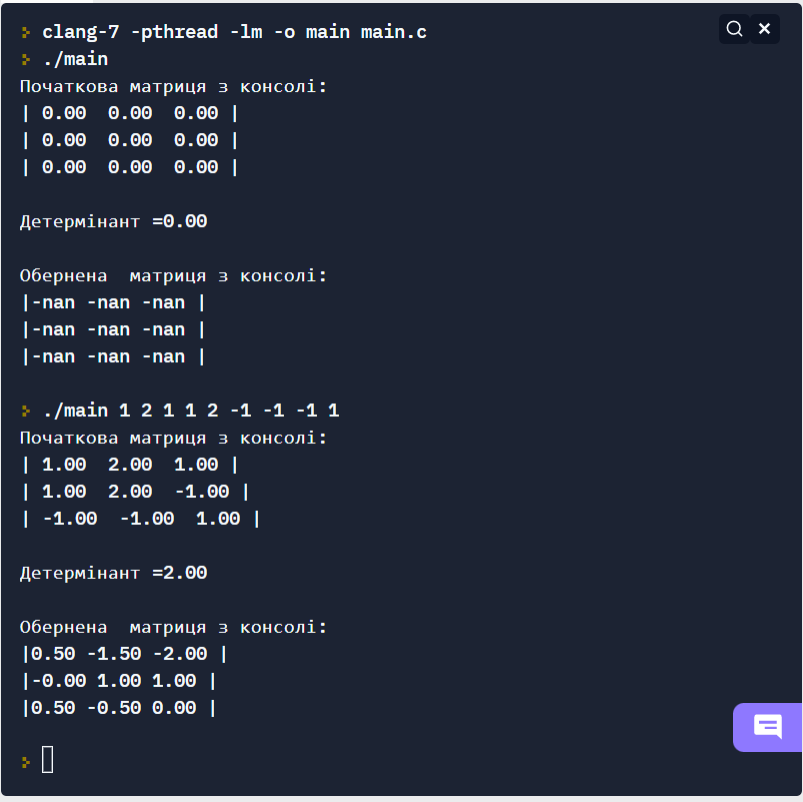
printf("\n");

}

printf("\n");

}

***Результат:***

******

***Посилання :***

[***https://replit.com/@VladKarkushievs/kr***](https://replit.com/@VladKarkushievs/kr)

***Висновки :***  під час виконання модульної контрольної роботи був написаний алгоритм для обчислення оберненої матриці та був виконаний вивід початкової і оберненої матриці .