

Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук

Департамент

Программной инженерии

Домашняя работа по
дисциплине
«Архитектура вычислительных систем»

Тема работы: Вариант 8. Используя формулы Крамера, найти решение системы линейных уравнений.
--

Выполнил: студент группы БПИ194
Гребенщиков М. М.

тел. +7 (922) 704 5875
e-mail адрес: mmgrebenschikov@edu.hse.ru

Преподаватель: Легалов Александр Иванович

Москва, 2020

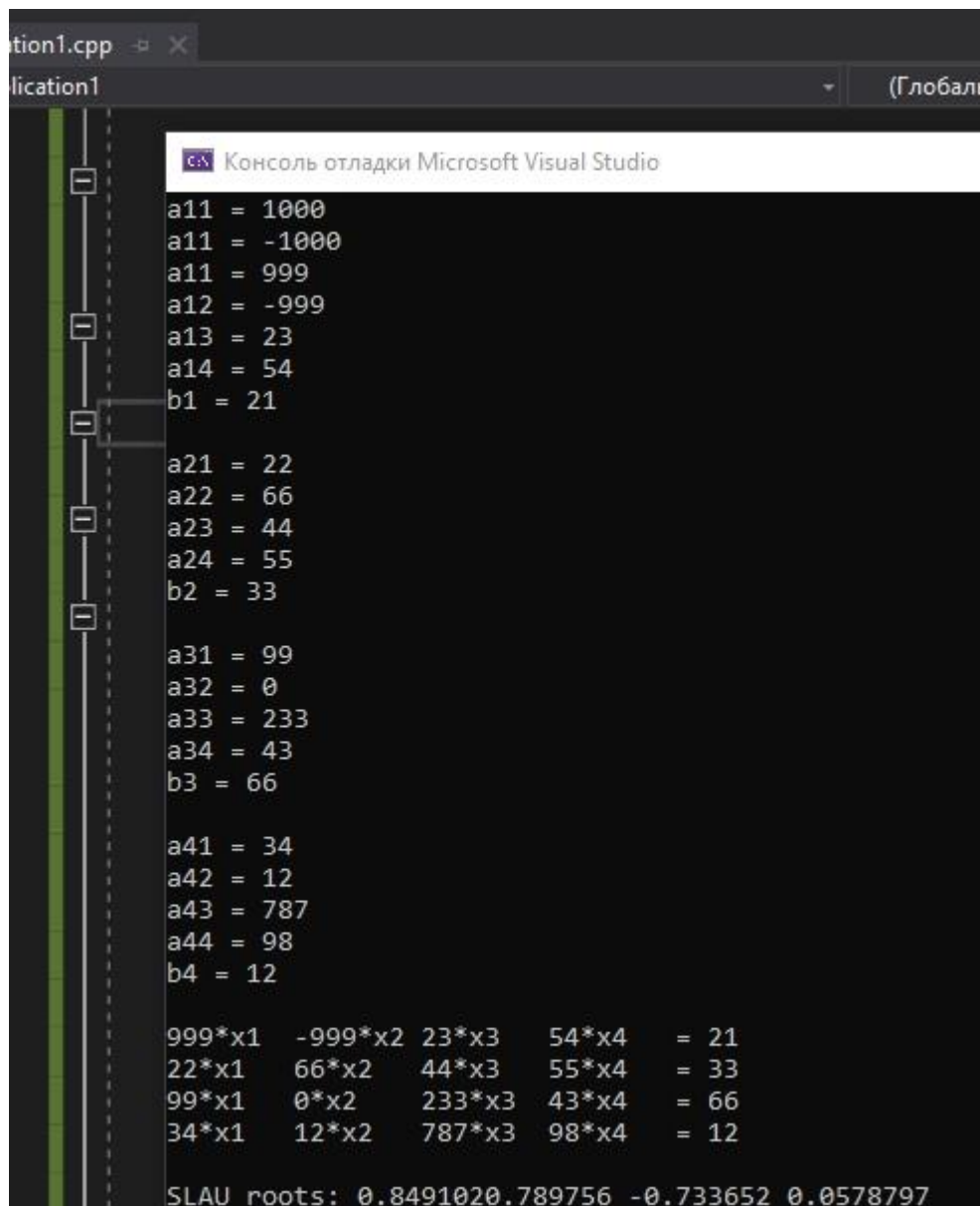
Структура работы

1. Kramer.cpp – содержит исходный код программы
2. inputData – тестовые наборы
3. Note.pdf – Отчёт

В коде программы находятся комментарии, описывавшие каждый шаг программы.

Тестирование программы

В качестве входных данных программа принимает коэффициенты СЛАУ из консоли. Пользователь вводит по 1 числу. Область допустимых значений: $[-999; 999]$. Предусмотрена обработка некорректных данных.



```
tion1.cpp x
lication1 (Глобаль

Консоль отладки Microsoft Visual Studio

a11 = 1000
a11 = -1000
a11 = 999
a12 = -999
a13 = 23
a14 = 54
b1 = 21

a21 = 22
a22 = 66
a23 = 44
a24 = 55
b2 = 33

a31 = 99
a32 = 0
a33 = 233
a34 = 43
b3 = 66

a41 = 34
a42 = 12
a43 = 787
a44 = 98
b4 = 12

999*x1 -999*x2 23*x3 54*x4 = 21
22*x1 66*x2 44*x3 55*x4 = 33
99*x1 0*x2 233*x3 43*x4 = 66
34*x1 12*x2 787*x3 98*x4 = 12

SLAU roots: 0.849102 0.789756 -0.733652 0.0578797
```

Test1

```
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
Application
Консоль отладки Microsoft Visual Studio
a11 = fsdf
a11 = asd
a11 = qwe
a11 = `q23
a11 = sfd234
a11 = sfd232
a11 = 90
a12 = 23
a13 = 43
a14 = 5
b1 = 1

a21 = 2
a22 = 9
a23 = -3
a24 = -45
b2 = 900

a31 = -45
a32 = 32
a33 = 54
a34 = 65
b3 = 2

a41 = 1
a42 = 78
a43 = 4
a44 = 65
b4 = 23

90*x1  23*x2  43*x3  5*x4  = 1
2*x1   9*x2   -3*x3  -45*x4 = 900
-45*x1 32*x2  54*x3  65*x4 = 2
1*x1   78*x2  4*x3   65*x4 = 23

SLAU roots: -6.32947 14.83617.40608 -17.8078

118
}
Консоль отладки Microsoft Visual Studio
a11 = 1
a12 = 1
a13 = 1
a14 = 1
b1 = 1

a21 = 2
a22 = 2
a23 = 2
a24 = 2
b2 = 2

a31 = 3
a32 = 3
a33 = 3
a34 = 3
b3 = 3

a41 = 4
a42 = 4
a43 = 4
a44 = 4
b4 = 4

1*x1  1*x2  1*x3  1*x4  = 1
2*x1  2*x2  2*x3  2*x4  = 2
3*x1  3*x2  3*x3  3*x4  = 3
4*x1  4*x2  4*x3  4*x4  = 4

Main determinant is 0. Can't use Kramer method
```

Test2

Test3

Алгоритм работы программы

1. Считывание коэффициентов СЛАУ
2. Вычисление основного определителя
3. Проверка определителя на равенство 0
4. Запуск 4 поток вычисления корней
5. Каждый поток выводит подсчитанный x_i на экран

Список литературы

1. https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9A%D1%80%D0%B0%D0%BC%D0%B5%D1%80%D0%B0

Приложение. Код программы

```
#include <iostream>
#include <fstream>
#include <thread>
#include <stdlib.h>
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <string>

//Метод вывода СЛАУ на экран
void printSLAU(int arc[4][5]) {
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 5; j++) {
            if (j == 4)
                std::cout << "=" << arc[i][j] << std::endl;
            else
                std::cout << arc[i][j] << "*x" << j + 1 << "\t";
        }
    }
}

//Вычисление определителя 3 на 3
long long calcDetThreeXThree(int arc[3][3]) {
    long long det = arc[0][0] * arc[1][1] * arc[2][2] + arc[0][1] * arc[1][2] * arc[2][0]
+ arc[1][0] * arc[2][1] * arc[0][2];
    det -= (arc[0][2] * arc[1][1] * arc[2][0] + arc[1][0] * arc[0][1] * arc[2][2] +
arc[0][0] * arc[2][1] * arc[1][2]);
    return det;
}

//Вычисление определителя 4 на 4
long long calcDetFourXFour(int arc[4][5]) {
    long long det = 0;
    int minor[3][3];

    //Разложение матрицы по первой строке
    for (int i = 0; i < 4; i++) {
        for (int j = 1; j < 4; j++)
        {
            int ind = 0;
            //Составление минора
            for (int k = 0; k < 4; k++)
            {
                if (k == i)
                    continue;

                minor[j - 1][ind] = arc[j][k];
                ind++;
            }
        }
        //Вычисление определителя минора
```

```

        long long minorDet = calcDetThreeXThree(minor);
        if (i % 2 == 1)
            minorDet *= -1;
        det += (long long)arc[0][i] * minorDet;
    }
    return det;
}

void calcRoot(int rootNum, int arc[4][5], long long mainDet) {
    //Замена столбца коэффициентов свободными членами
    int arrc[4][5];
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++) {
            if (j == rootNum)
            {
                arrc[i][j] = arc[i][4];
            }
            else
            {
                arrc[i][j] = arc[i][j];
            }
        }
    }

    //Вычисление определителя
    long long det = calcDetFourXFour(arrc);
    //Вывод и вычисление корня
    std::cout << (double)det / (double)mainDet << " ";
}

int main()
{
    int arc[4][5];

    //Считывание входных данных из консоли
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            while (true)
            {
                try
                {
                    std::string s;
                    if (j != 4)
                        std::cout << "a" << i + 1 << j + 1 << " = ";
                    else
                        std::cout << "b" << i + 1 << " = ";

                    std::cin >> s;
                    arc[i][j] = std::stoi(s);
                    //Обработка некорректных входных данных
                    if (arc[i][j] >= 1000 || arc[i][j] <= -1000)
                        continue;
                    break;
                }
                catch (std::invalid_argument) {
                    continue;
                }
            }
        }
    }
}

```

```

        }
    }
    std::cout << std::endl;
}

std::ifstream in("input/");

//Вывод СЛАУ на экран
printSLAU(arc);

//Высечение детерминанта коэффициентов
long long mainDet = calcDetFourXFour(arc);

//Если детерминант 0 - Крамера использовать нельзя
if (mainDet == 0)
{
    std::cout << std::endl << "Main determinant is 0. Can't use Kramer method";
    return 0;
}

std::cout << std::endl << "SLAU roots: ";

//Запуск поток для вычисления корней
std::thread tA(calcRoot, 0, arc, mainDet);
std::thread tB(calcRoot, 1, arc, mainDet);
std::thread tC(calcRoot, 2, arc, mainDet);
std::thread tD(calcRoot, 3, arc, mainDet);

tA.join();
tB.join();
tC.join();
tD.join();
}

```