Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №1

по курсу «Исследование операций»

***Симплекс-метод***

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Коржавина А.С./

Киров 2022

1. Цель

Закрепить на практике знания о симплексном методе решения задач линейного программирования и получить навыки его программной реализации.

2. Задание

написать программу, реализующую симплексный метод либо одну из его модификаций – метод искусственного базиса, двойственный симплекс и т. д.

Необходимо рассмотреть все возможные случаи, например, когда задача не имеет решений, имеет множество решений и т.д.

3. Описание симплекс метода

Его суть заключается в том, что вместо того, чтобы искать базис среди имеющихся основных и дополнительных переменных, ввести так называемые искусственные переменные, которые сформируют начальный базис. Возможно, звучит сложно и непонятно, но сейчас мы всё объясним.

**Подготовительный этап**

Аналогично базовому симплекс-методу для всех ограничений с неравенством вводятся дополнительные переменные, причём для ограничений с ≥ они берутся с коэффициентом -1, а для ограничений с ≤ с коэффициентом 1. Ограничения с равенством остаются без изменений. Если свободный коэффициент какого-либо из ограничений меньше нуля, то такое ограничение умножается на -1 (знак неравенства при этом меняется на противоположный). После этого приступают к поиску базиса.

### Формирование начального базиса

Для того, чтобы сформировать начальный базис в первую очередь можно поискать столбец, у которого одно значение равно единице, а все значения остальные значения равны нулю, и сделать соответствующую переменную базисной для этой строки. Однако такое случается довольно редко, поэтому проще сразу перейти к следующему пункту. Для всех ограничений, не имеющих базисной переменной, добавляем искусственную переменную с коэффициентом 1. В целевую функцию добавляем эту же переменную с коэффициентов -M, если ищется максимум или с коэффициентом M, если ищется минимум. M всего лишь является очень большим числом.

### Расчёт дельт и проверка плана на оптимальность

После того, как начальный базис сформирован необходимо вычислить дельты. Дельты вычисляются полностью аналогично базовому методу: Δi = ce1·a1i + ce2·a2i + ... + cem·ami - ci. Единственным отличием будет тот факт, что результат может содержать значения с M. Когда дельты будут получены необходимо проверить текущий опорный план на оптимальность (см. проверку плана на оптимальность в базовом симплекс-методе). Если план оптимален, то алгоритм завершает свою работу, иначе формирует более оптимальное решение и повторяет процесс.

4. Листинг программы

#include <iostream>

using namespace std;

void vvod(double\*\* matrix, int n, int m)

{

for (int i = 0; i < n; i++)

{

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

{

if (j < m - n)

{

cout << "Ведите а[" << i << "," << j << "]: ";

cin >> matrix[i][j];

}

else

{

if (i == j - n)

{

matrix[i][j] = 1;

}

else

{

matrix[i][j] = 0;

}

}

}

}

}

void out(double\*\* matrix, int n, int m)

{

for (int i = 0; i < n; i++)

{

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

{

cout << matrix[i][j] << " ";

}

cout << '\n';

}

}

void divided(int str, int row, double\*\* matrix, int m)

{

double l = matrix[str][row];

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

{

matrix[str][j] = matrix[str][j] / l;

}

}

void guass(double\*\* matrix, int n, int str, int row, int m)

{

double p = 0;

for (int i = 0; i < n; i++)

{

p = matrix[i][row];

if (i != str)

{

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

{

matrix[i][j] = matrix[i][j] - matrix[str][j] \* p;

}

}

}

}

int main()

{

setlocale(LC\_ALL, "rus");

cout << "Введите кол-во производимой продукции (кол-во столбцов) : ";

int m;

cin >> m;

cout << "Введите кол-во ограничений (кол-во строк) : ";

int n;

cin >> n;

int s = m + n;

double\*\* matrix;

matrix = new double\* [n]; //каждой строке

for (int i = 0; i < n; i++)

{

matrix[i] = new double[s]; //по m столбцов

}

vvod(matrix, n, s);

out(matrix, n, s);

int str;

int row;

cout << "Введите строку: ";

cin >> str;

cout << "Введите столбец: ";

cin >> row;

divided(str, row, matrix, n);

out(matrix, n, s);

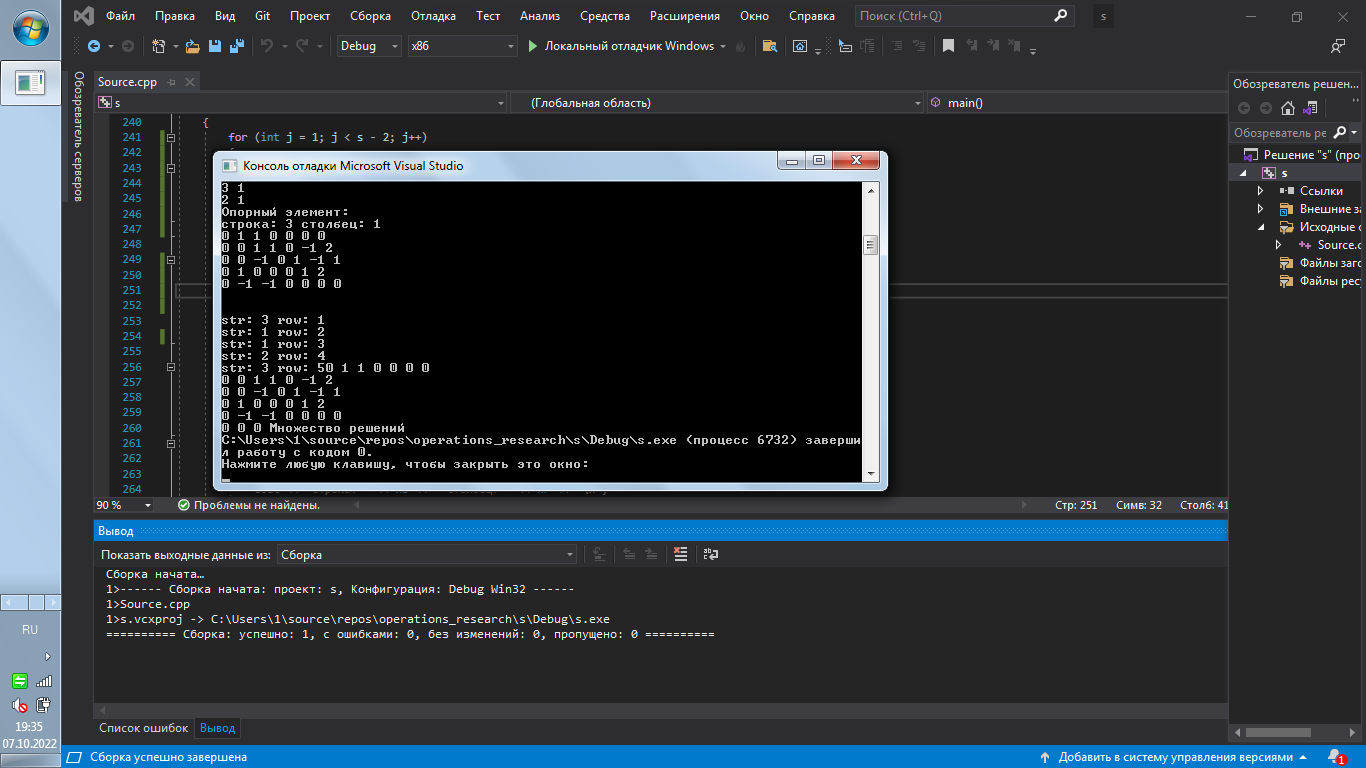
guass(matrix, n, str, row, s);

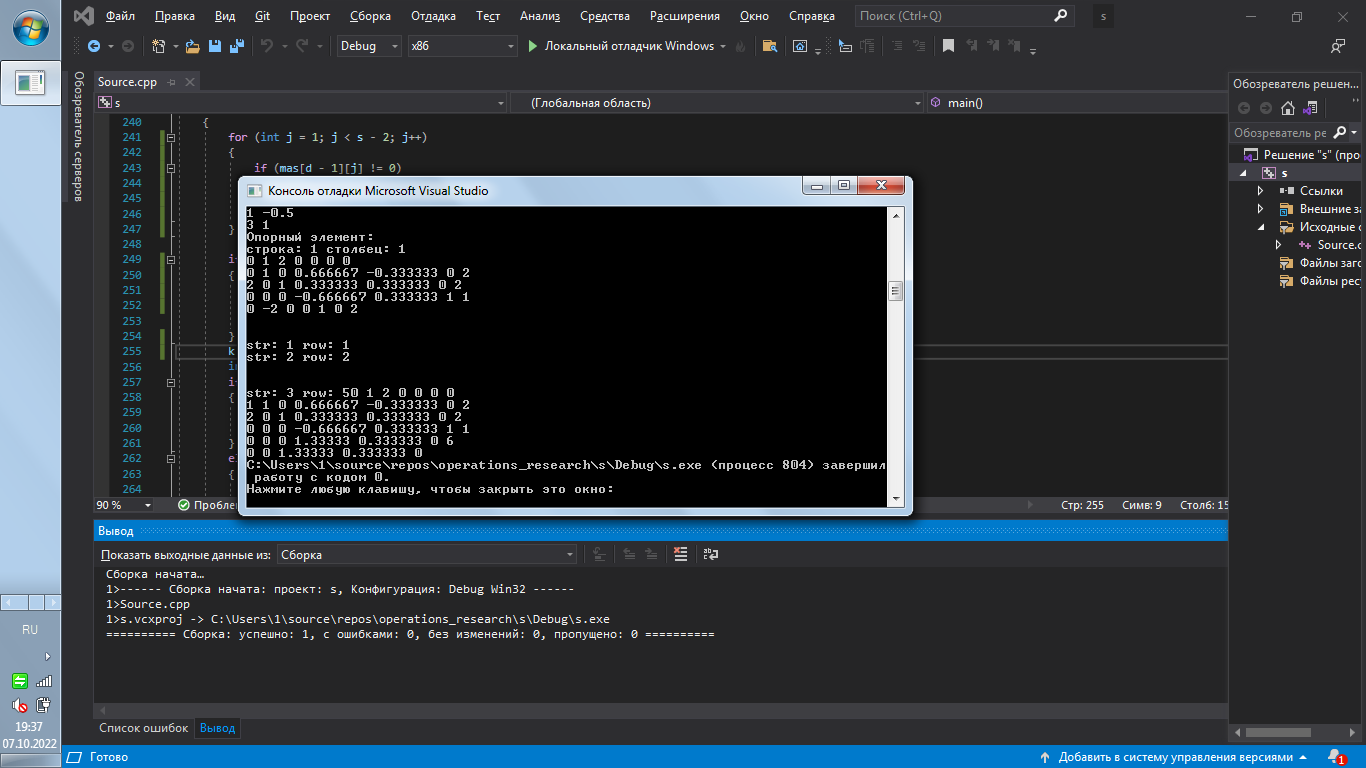
out(matrix, n, s);

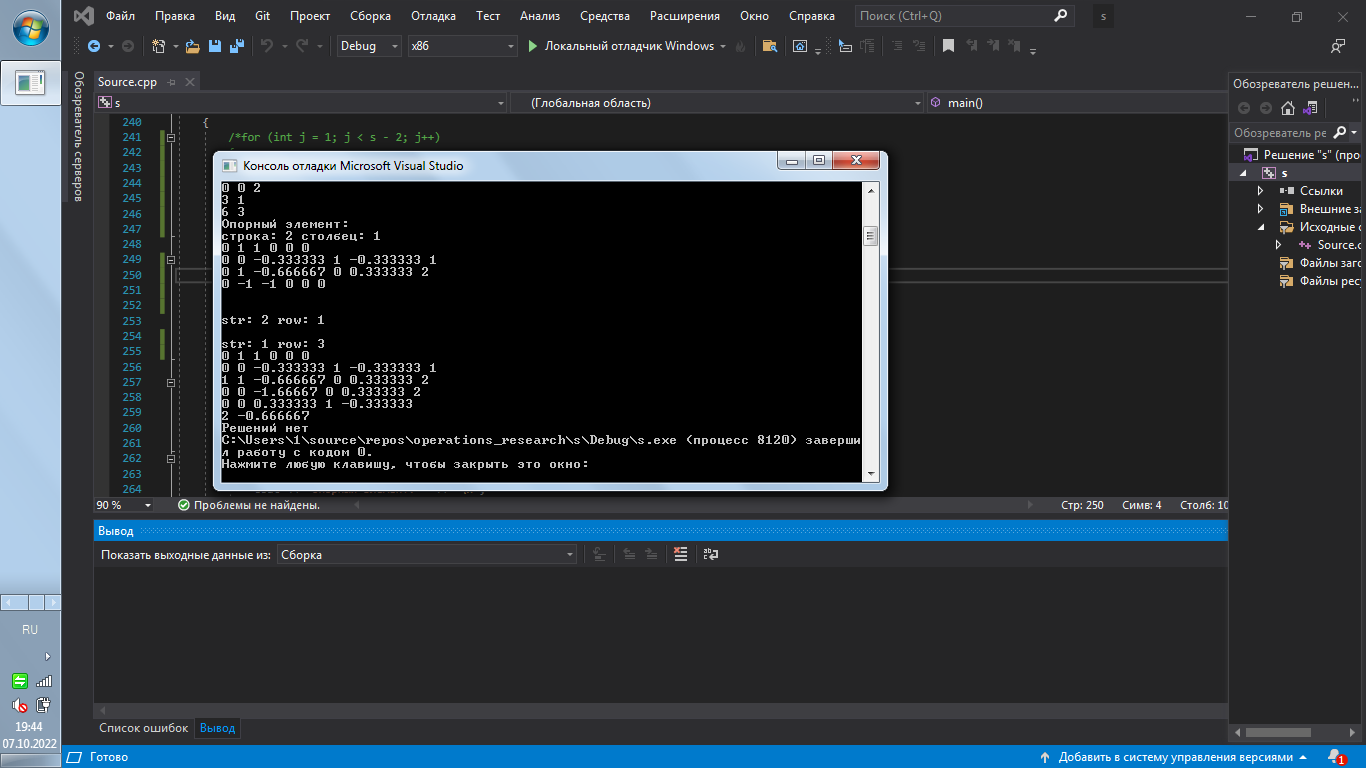
return 0;

}

5. Экранные формы







6. Вывод

В ходе выполнения лабораторной работы были закреплены на практике знания о симплексном методе решения задач линейного программирования и получены навыки его программной реализации