Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Факультет информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

Кафедра прикладной математики и кибернетики

**Лабораторная работа №3  
 Алгоритмы и вычислительные методы оптимизации**

**Нахождение начального опорного плана транспортной задачи методом северо-западного угла**

Выполнил:

студент гр. ИП-711 \_\_\_\_\_\_\_\_\_\_\_\_\_/Мартасов И.О/

ФИО студента

«14» мая 2020 г.

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_/Галкина М.Ю./

ФИО преподавателя

«14» мая 2020 г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2020 г.

**Содержание**

1. Постановка задачи …..........................................................................................3

2. Код программы……............................................................................................4

3. Результаты тестирования…........………………………………………………8

**Постановка задачи**

Написать программу, находящую начальный опорный план транспортной

задачи методом северо-западного угла.

Матрицу тарифов, запасы поставщиков и потребности потребителей

вводить из файла.

Программа должна работать как с открытой, так и закрытой моделью

транспортной задачи. Предусмотреть программное нахождение вырожденного плана.

Вывести распределение перевозок и затраты. Для тестирования

использовать несколько заданий из практических занятий.

**Код программы**

#include <iostream>

#include <fstream>

using namespace std;

int min(int a, int b)

{

return (a < b ? a : b);

}

void print(int \*\*a, int \*\*traffic, int \*dop, bool issklad, bool iseat, int \*sklad, int \*eat, int n, int m)

{

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

{

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

{

cout << a[i][j] << "/" << traffic[i][j] << "\t";

}

if (issklad)

{

cout << dop[i] << "/0\t";

}

cout << sklad[i] << "\t";

cout << endl;

}

if (iseat)

{

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

{

cout << dop[i] << "/0\t";

}

cout << dop[m] << "\t";

cout << endl;

}

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

{

cout << eat[i] << "\t";

}

if (issklad)

{

cout << dop[n] << "\t";

}

cout << endl;

cout << endl;

}

int main()

{

ifstream fin("in.txt");

int n;

int m;

bool issklad = 0;

bool iseat = 0;

bool isvyr = 0;

int x = 0;

int y = 0;

int obsklad = 0;

int obeat = 0;

int\* dop;

int sum = 0;

int count = 0;

fin >> n >> m;

int\* sklad = new int[n];

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

{

fin >> sklad[i];

}

int\* eat = new int[m];

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

{

fin >> eat[i];

}

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

{

obsklad += sklad[i];

}

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

{

obeat += eat[i];

}

if (obsklad < obeat)

{

dop = new int[m + 1];

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

{

dop[i] = 0;

}

dop[m] = obeat - obsklad;

iseat = 1;

}

else if (obsklad > obeat)

{

dop = new int[n + 1];

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

{

dop[i] = 0;

}

dop[n] = obsklad - obeat;

issklad = 1;

}

int \*\*traffic = new int\*[n];

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

{

traffic[i] = new int[m];

}

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

{

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

{

fin >> traffic[i][j];

}

};

int \*\*a = new int\*[n];

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

{

a[i] = new int[m];

}

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

{

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

{

a[i][j] = 0;

}

}

print(a, traffic, dop, issklad, iseat, sklad, eat, n, m);

while (1)

{

if (iseat && x == n)

{

count = min(dop[m], eat[y]);

dop[y] = count;

dop[m] -= count;

eat[y] -= count;

if (eat[y] == 0 && dop[m] == 0)

{

break;

}

if (eat[y] == 0)

{

y++;

}

}

else if (issklad && y == m)

{

count = min(sklad[x], dop[n]);

dop[x] = count;

sklad[x] -= count;

dop[n] -= count;

if (sklad[x] == 0 && dop[n] == 0)

{

break;

}

if (sklad[x] == 0)

{

x++;

}

}

else

{

count = min(sklad[x], eat[y]);

a[x][y] = count;

sklad[x] -= count;

eat[y] -= count;

if (sklad[x] == 0 && eat[y] == 0)

{

isvyr = 1;

}

if (sklad[x] == 0)

{

x++;

if (x == n && !iseat)

{

x--;

}

}

if (eat[y] == 0)

{

y++;

if (y == m && !issklad)

{

y--;

}

}

if (sklad[x] == 0 && eat[y] == 0 && (x == n - 1 || y == m - 1))

{

break;

}

}

print(a, traffic, dop, issklad, iseat, sklad, eat, n, m);

}

print(a, traffic, dop, issklad, iseat, sklad, eat, n, m);

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

{

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

{

sum += traffic[i][j] \* a[i][j];

}

}

if (isvyr)

{

cout << "plan vyrozdenny" << endl;

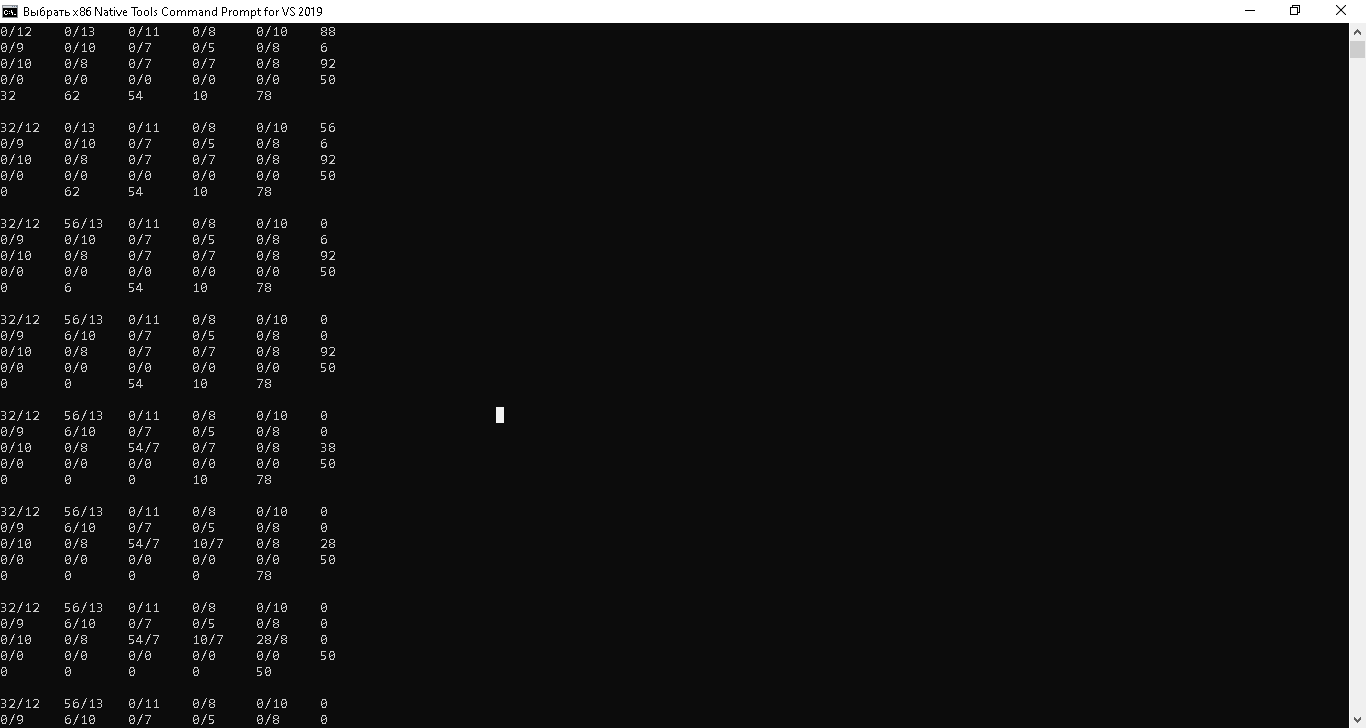
}

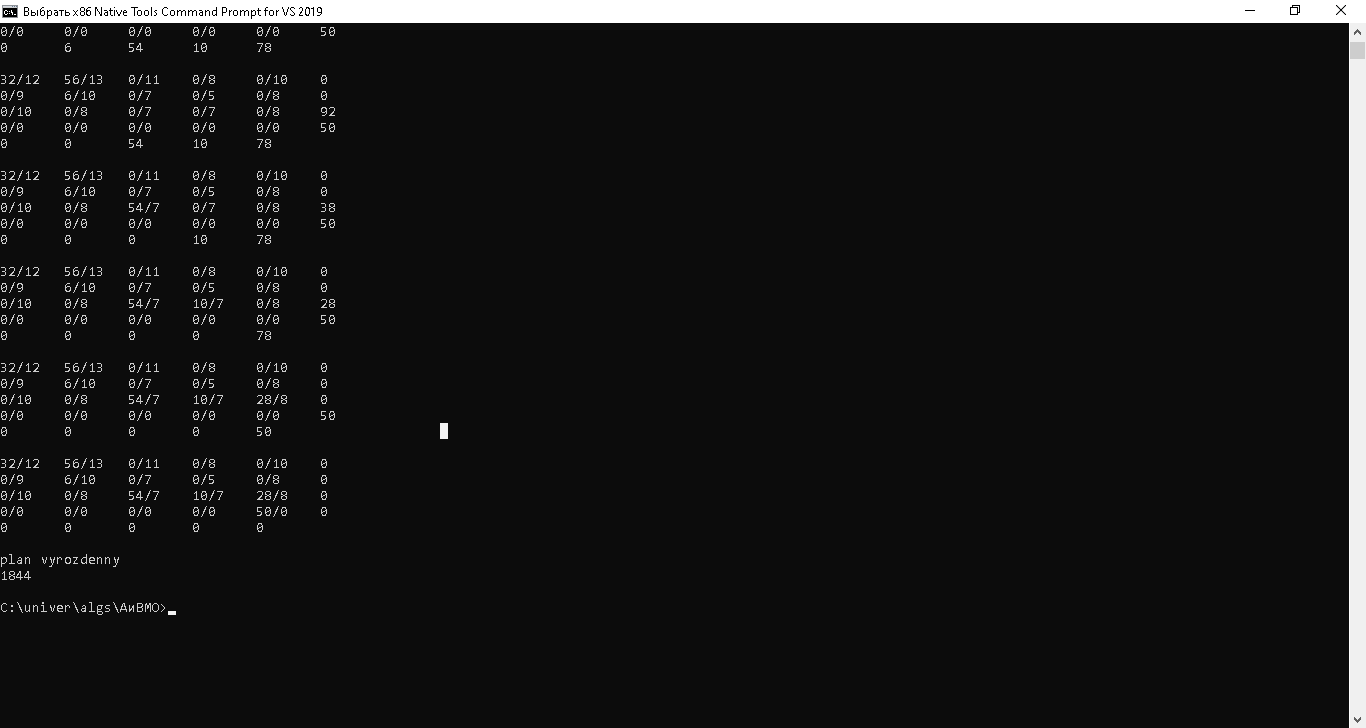
cout << sum << endl;

}

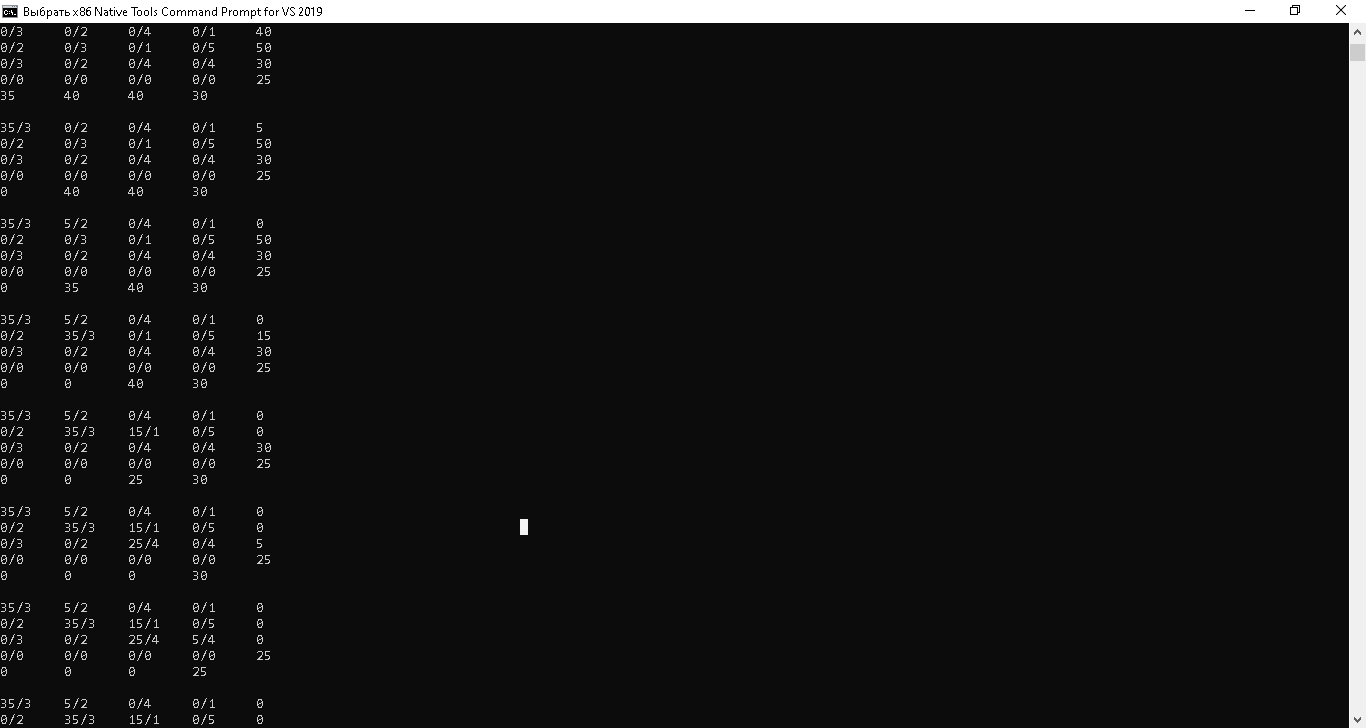
**Результаты тестирования**

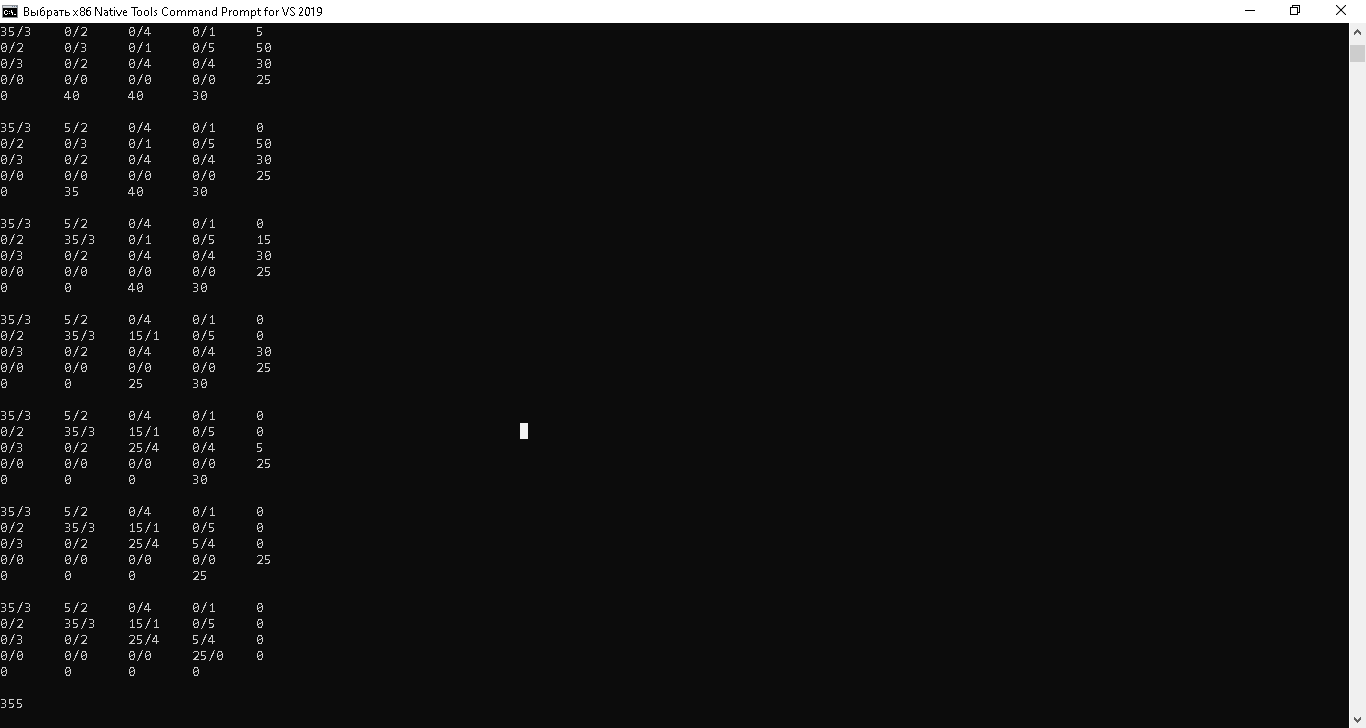
**1.Задание 1(а)**

****

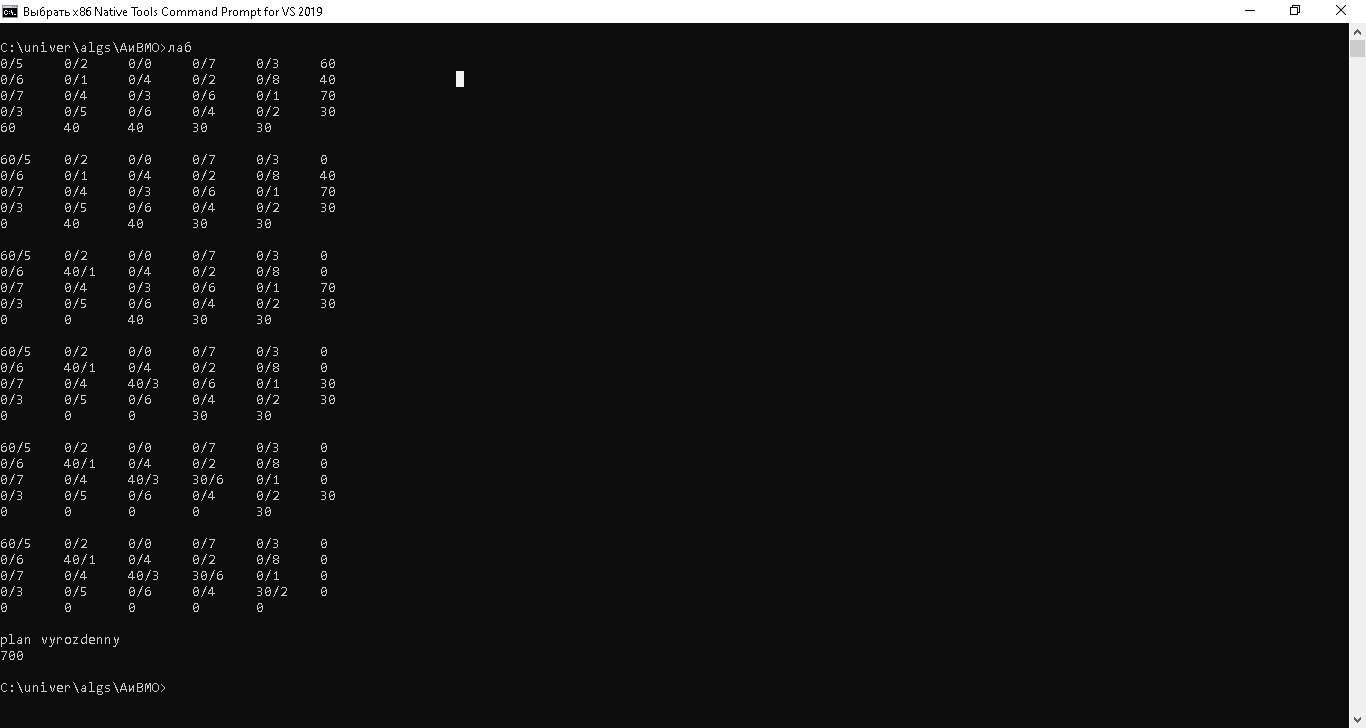
****

**2.Задание 2(з)**

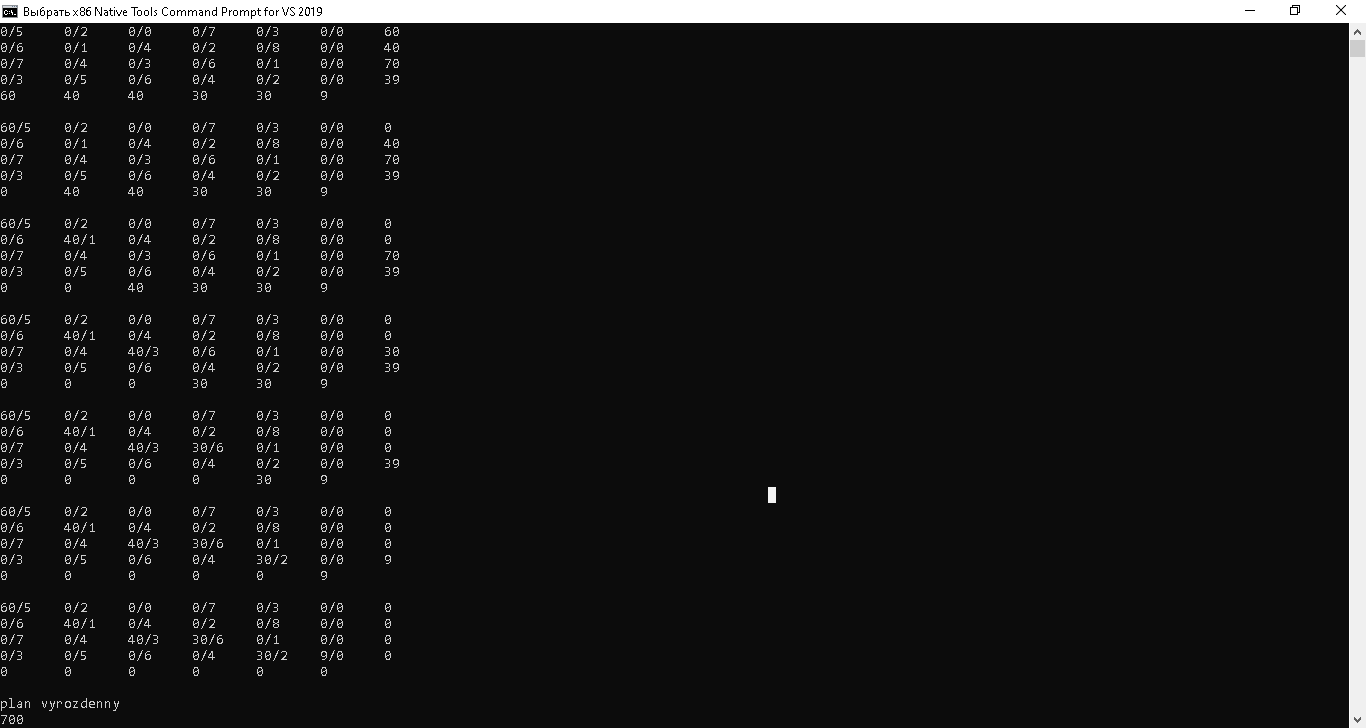
****

****

**3. Задание 2(г)**

****

**4.Задание 2(г) – изменено для добавления дополнительного столбца**

****