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

**Курсовая работа по дисциплине  
 Алгоритмы и вычислительные методы оптимизации**

**Симплекс-метод**

Вариант 10

Выполнила:

студентка гр.ИП-813 \_\_\_\_\_\_\_\_\_\_\_\_/Захарова К.Ю./

ФИО студента

«07» апреля 2021 г.

Проверила

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

ФИО преподавателя

«19» мая 2021 г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2021 г.

**Содержание**

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

2. Решение задачи……............................................................................................4

2.1.Задание 1……………………………………………..…………………4

2.2.Задание 2……………………………………………………..…………4

2.3.Задание 3………..………………………………………………………6

2.4.Задание 4…..……………………………………………………………7

3. Приложение. Код программы.....………………………………………………8

**Постановка задачи**

**Решение задачи линейного программирования, теория двойственности**

1. Перейти к канонической форме записи задачи линейного программирования.
2. Написать программу, решающую задачу линейного программирования в канонической форме (с выводом всех промежуточных таблиц):

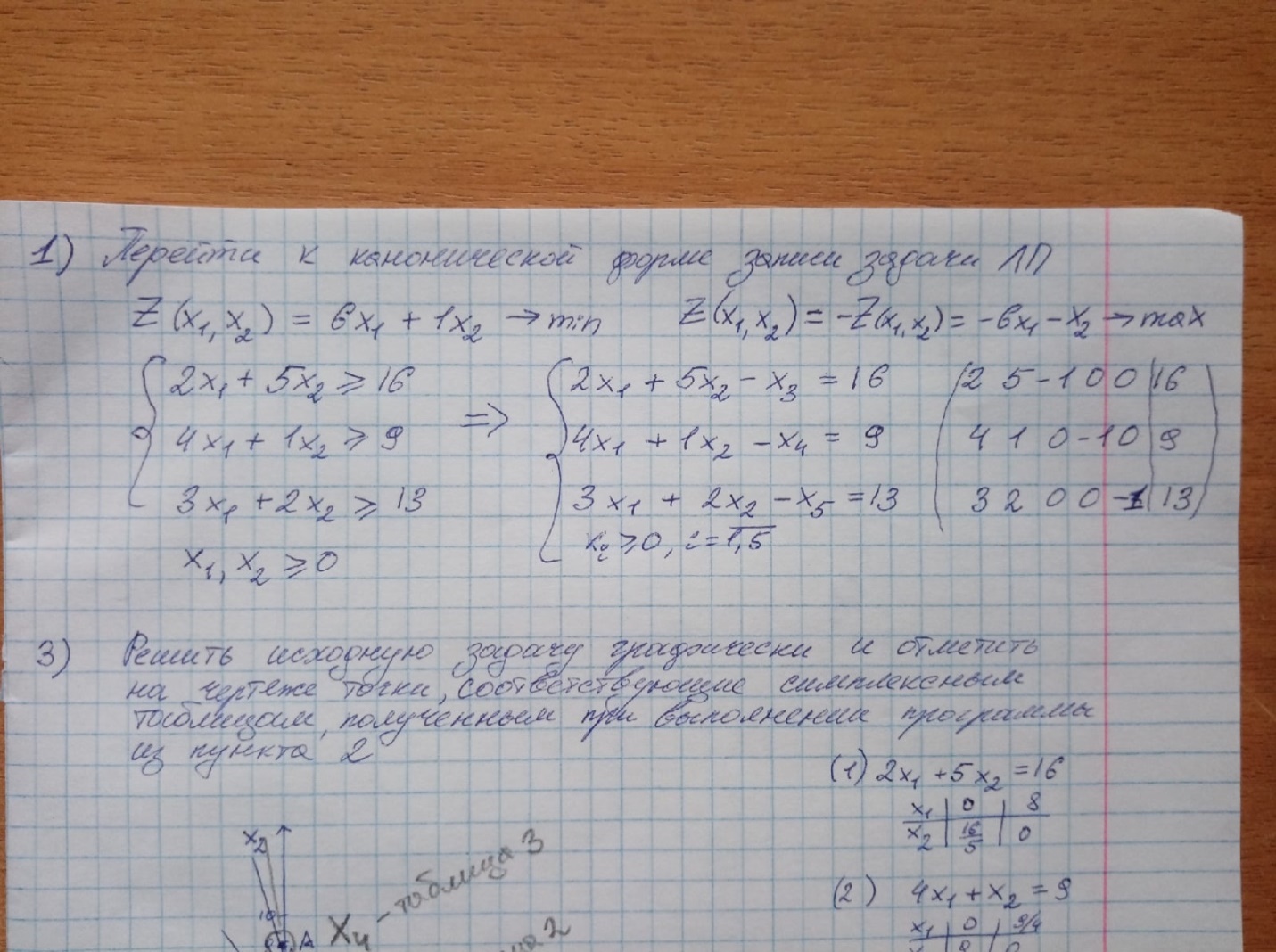
- симплекс-методом, используя в качестве начальной угловой точки опорное решение с указанными в задании базисными переменными, найденное методом Жордана - Гаусса.

1. Решить исходную задачу графически и отметить на чертеже точки, соответствующие симплексным таблицам, полученным при выполнении программы из п.2.
2. Составить двойственную задачу к исходной и найти ее решение на основании теоремы равновесия.

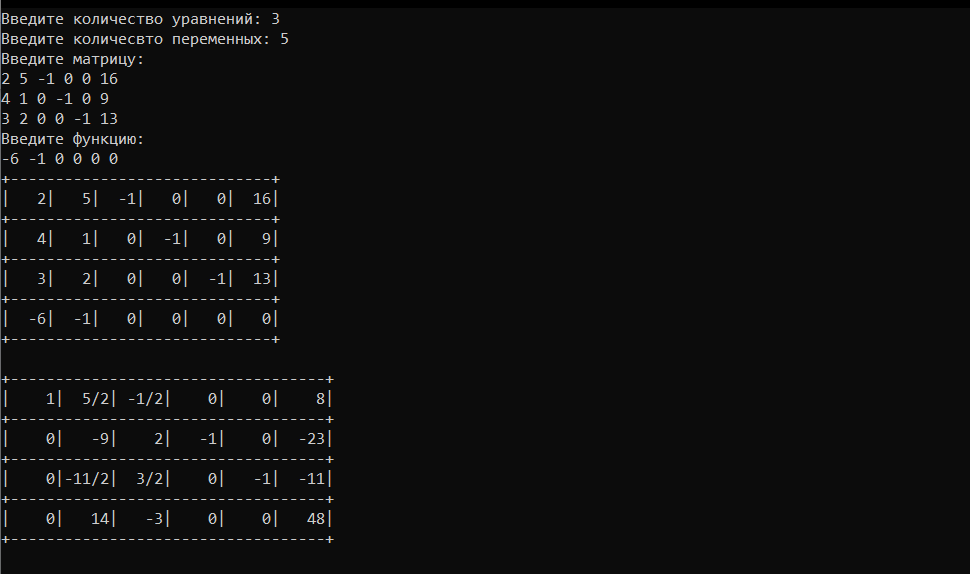
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Номер варианта** | **a** | **b** | **c** | **a1** | **b1** | **c1** | **a2** | **b2** | **c2** | **p1** | **p2** | **Метод решения задачи** |
| **10** | 16 | 9 | 13 | 2 | 4 | 3 | 5 | 1 | 2 | 6 | 1 | 1  Базисные переменные |

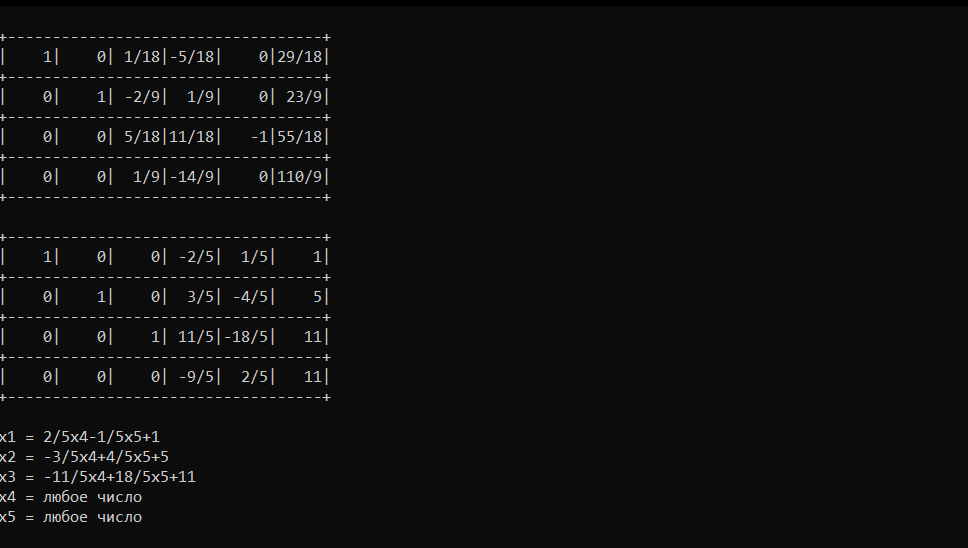
**Решение задачи**

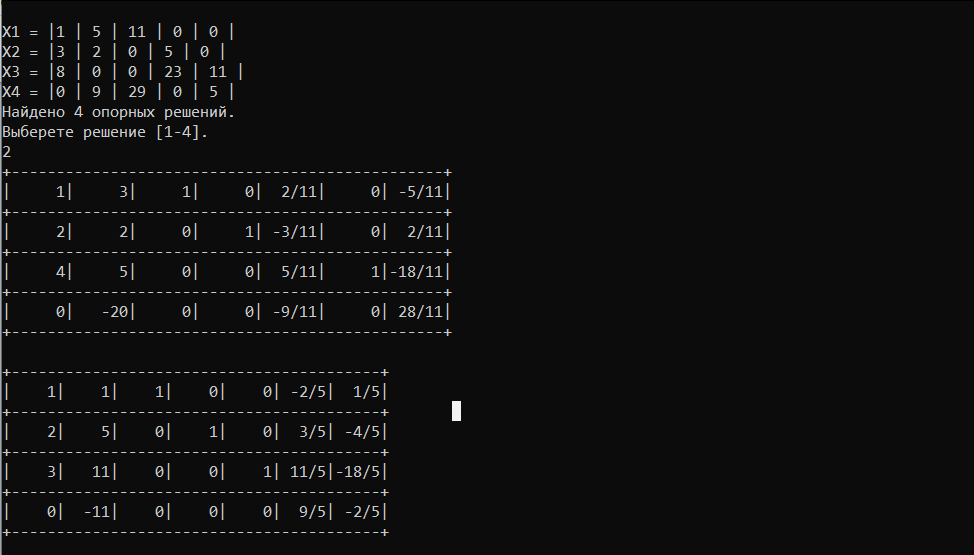
**Задание 1**

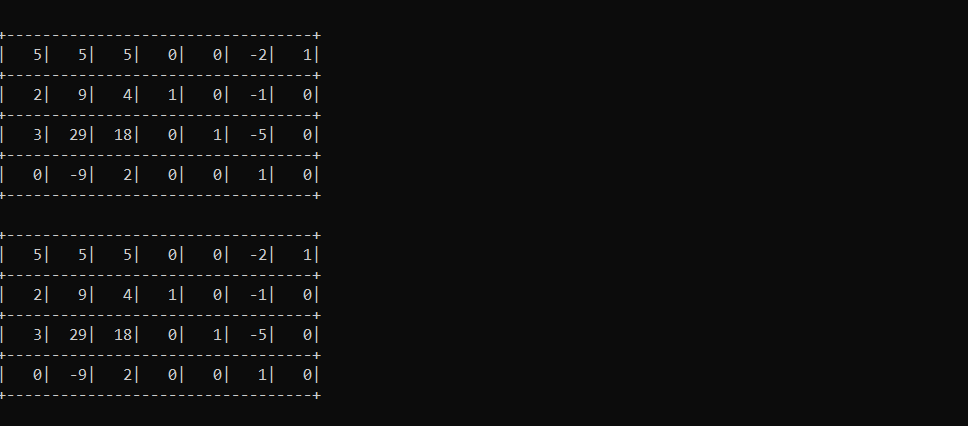


**Задание 2**

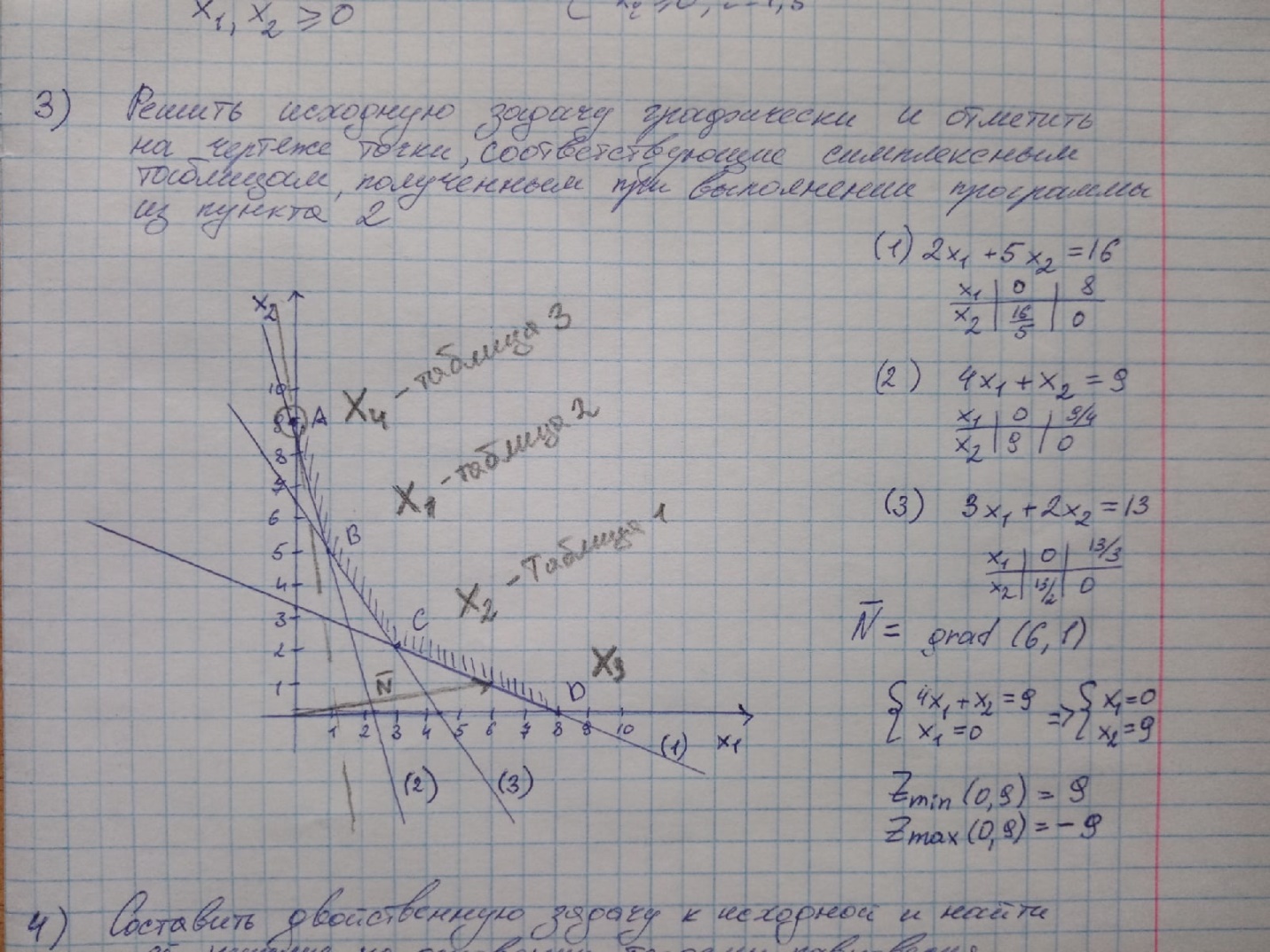




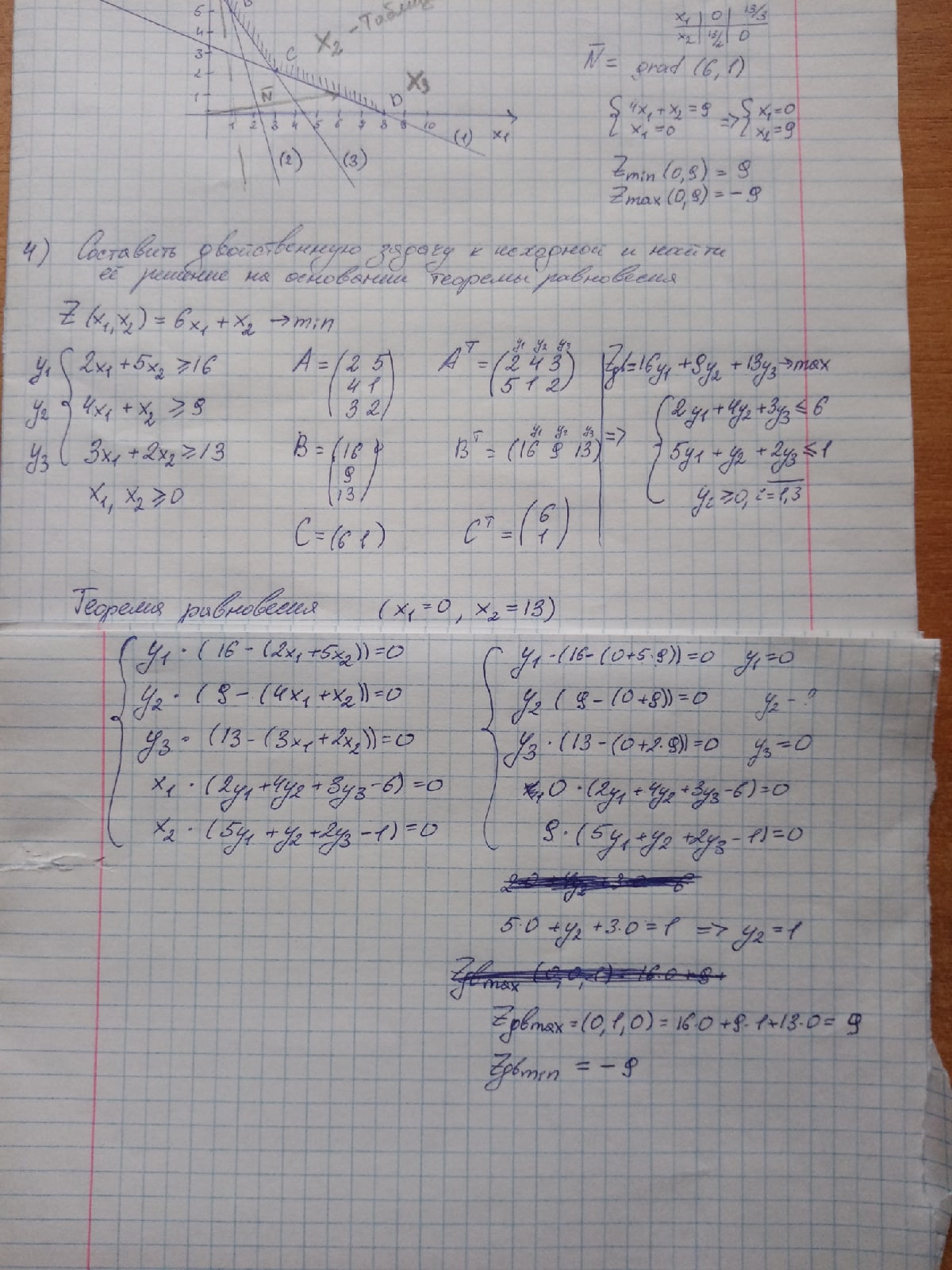




**Задание 3**



**Задание 4**



**Приложение. Код программы**

#include <iostream>

#include <windows.h>

#include <vector>

#include <string>

#include <algorithm>

#include <vector>

#include "bprinter/table\_printer.h"

using namespace std;

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);//для ввода и вывода

#pragma region Rational

class Rational

{

private:

long long numerator, denominator;

long long Nod(long long x, long long y) {

if (y == 0)

return x;

return Nod(y, x % y);

}

public:

void Cancel() {

if (this->Numerator == 0) {

this->Denominator = 1;

return;

}//для определения знака

long long tmp = Nod(numerator, denominator);

numerator /= tmp;

denominator /= tmp;

if (denominator < 0) {

numerator \*= -1;

denominator \*= -1;

}

}

Rational() {

this->numerator = 0;

this->denominator = 1;

}

Rational(long long number) {

this->numerator = number;

this->denominator = 1;

}

Rational(long long numerator, long long denominator) {

if ((numerator < 0 && denominator < 0) || denominator < 0) {

this->numerator = -1 \* numerator;

this->denominator = -1 \* denominator;

}

else {

this->numerator = numerator;

this->denominator = denominator;

}

this->Cancel();

}

Rational(const Rational& other) {

this->numerator = other.numerator;

this->denominator = other.denominator;

this->Cancel();

}

\_\_declspec(property(get = getNumerator, put = putNumerator)) long long const Numerator;

long long getNumerator() const

{

return numerator;

}

void putNumerator(long long value)

{

numerator = value;

}

\_\_declspec(property(get = getDenominator, put = putDenominator)) long long const Denominator;

long long getDenominator() const

{

return denominator;

}

void putDenominator(long long value)

{

denominator = value;

}

Rational& operator=(const Rational& other) {

this->numerator = other.numerator;

this->denominator = other.denominator;

this->Cancel();

return \*this;

}

Rational& operator+=(const Rational& other) {

this->numerator = this->numerator \* other.denominator + this->denominator \* other.numerator;

this->denominator = this->denominator \* other.denominator;

this->Cancel();

return \*this;

}

Rational& operator-=(const Rational& other) {

this->numerator = this->numerator \* other.denominator - this->denominator \* other.numerator;

this->denominator = this->denominator \* other.denominator;

this->Cancel();

return \*this;

}

Rational& operator\*=(const Rational& other) {

this->numerator \*= other.numerator;

this->denominator \*= other.denominator;

this->Cancel();

return \*this;

}

Rational& operator/=(const Rational& other) {

if (other.Numerator == 0) {

this->numerator = 0;

this->denominator = 1;

return \*this;

}

this->numerator \*= other.denominator;

this->denominator \*= other.numerator;

this->Cancel();

return \*this;

}

};

Rational operator-(const Rational& a) { return Rational(-a.Numerator, a.Denominator); }

Rational operator+(const Rational& a) { return a; };

Rational operator+(const Rational& a, const Rational& b) {

Rational tmp(a.Numerator \* b.Denominator + a.Denominator \* b.Numerator, a.Denominator \* b.Denominator);

tmp.Cancel();

return tmp;

}

Rational operator-(const Rational& a, const Rational& b) {

Rational tmp(a.Numerator \* b.Denominator - a.Denominator \* b.Numerator, a.Denominator \* b.Denominator);

tmp.Cancel();

return tmp;

}

Rational operator\*(const Rational& a, const Rational& b) {

Rational tmp(a.Numerator \* b.Numerator, a.Denominator \* b.Denominator);

tmp.Cancel();

return tmp;

}

Rational operator/(const Rational& a, const Rational& b) {

Rational tmp(a.Numerator \* b.Denominator, a.Denominator \* b.Numerator);

tmp.Cancel();

return tmp;

}

bool operator==(const Rational& a, const Rational& b) {

if (a.Numerator == b.Numerator && a.Denominator == b.Denominator) return true;

return false;

}

bool operator!=(const Rational& a, const Rational& b) {

if (a.Numerator == b.Numerator && a.Denominator == b.Denominator) return false;

return true;

}

bool operator<(const Rational& a, const Rational& b) {

if (a.Numerator \* b.Denominator < a.Denominator \* b.Numerator) return true;

return false;

}

bool operator<=(const Rational& a, const Rational& b) {

if (a.Numerator \* b.Denominator <= a.Denominator \* b.Numerator) return true;

return false;

}

bool operator>(const Rational& a, const Rational& b) {

if (a.Numerator \* b.Denominator > a.Denominator \* b.Numerator) return true;

return false;

}

bool operator>=(const Rational& a, const Rational& b) {

if (a.Numerator \* b.Denominator >= a.Denominator \* b.Numerator) return true;

return false;

}

std::ostream& operator<<(std::ostream& stream, const Rational& num) {

if (num.Denominator == 1) stream << num.Numerator;

else stream << num.Numerator << "/" << std::noshowpos << num.Denominator;

return stream;

}

std::istream& operator>>(std::istream& stream, Rational& num) {

string s;

stream >> s;

string delimiter = "/";

size\_t pos = 0;

string token;

vector<long long> vec;

while ((pos = s.find(delimiter)) != std::string::npos) {

token = s.substr(0, pos);

vec.push\_back(atoi(token.c\_str()));

s.erase(0, pos + delimiter.length());

}

vec.push\_back(atoi(s.c\_str()));

if (vec.size() > 2) exit(-4);

else if (vec.size() == 2) {

num.Numerator = vec[0];

num.Denominator = vec[1];

num.Cancel();

}

else if (vec.size() == 1) {

num.Numerator = vec[0];

num.Denominator = 1;

}

else {

exit(-1);

}

return stream;

}

#pragma endregion

/\* Главная функция MAIN \*/

void outVar(Rational a, bool showpos = false) {

if (showpos)

cout << std::showpos;

else

cout << std::noshowpos;

cout << a << std::noshowpos;

}

void output(Rational\*\* a, int M, int N) {

int len = 1;

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

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

if (a[i][j].Denominator == 1)

len = max(len, (int)to\_string(a[i][j].Numerator).length());

else

len = max(len, (int)to\_string(a[i][j].Numerator).length() + 1 +

(int)to\_string(a[i][j].Denominator).length());

}

}

bprinter::TablePrinter tp(&std::cout);

for (int i = 0; i < N; i++) tp.AddColumn("", max(len, 4));

tp.PrintFooter();

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

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

if (a[i][j].Denominator == 1) {

tp << a[i][j].Numerator;

}

else {

tp << to\_string(a[i][j].Numerator) + "/" + to\_string(a[i][j].Denominator);

}

}

tp.PrintFooter();

}

cout << endl;

}

void Simplex(Rational\*\* a, int M, int N) {

Rational\*\* sys = new Rational \* [M];

sys[0] = new Rational[M \* N];

for (int i = 1; i < M; i++) {

sys[i] = sys[i - 1] + N;

}

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

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

sys[i][j] = a[i][j];

}

}

int c = 0;//количество переменных меньше нуля

int minRow = 0, minCol = 0;

for (int i = 2; i < N; i++) {

if (sys[M - 1][i] < Rational(0)) c++;

if (sys[M - 1][i] < sys[M - 1][minCol])minCol = i;

}

output(sys, M, N);

while (c > 0) {

bool checkRow = 0;

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

if (sys[i][minCol] < Rational(0))continue;

else if (!checkRow) { minRow = i; checkRow = 1; }

if (sys[i][1] / sys[i][minCol] < sys[minRow][1] / sys[minRow][minCol])

{

minRow = i;

checkRow = 1;

}

}

if (!checkRow) { cout << "Функция не имеет решений"; return; }

sys[minRow][0] = Rational(minCol - 1);

for (int i = M - 1; i >= 0; i--) {

for (int j = N - 1; j > 0; j--) {

if (i != minRow && j != minCol) {

sys[i][j] = sys[i][j] - sys[minRow][j] \* sys[i][minCol] / sys[minRow][minCol];

}

}

}

Rational tmp = sys[minRow][minCol];

for (int i = N - 1; i > 0; i--) {

sys[minRow][i] /= tmp;

}

for (int i = M - 1; i >= 0; i--) {

if (i != minRow) sys[i][minCol] = Rational(0);

}

output(sys, M, N);

c = 0;

for (int i = 2; i < N; i++) {

if (sys[M - 1][i] < Rational(0)) c++;

if (sys[M - 1][i] < sys[M - 1][minCol])minCol = i;

}

}

output(sys, M, N);

}

void basis(Rational\*\* a, int MS, int N, int rank) {

int M = MS - 1;

int\* q = new int[rank];

vector<Rational\*\*> bases;

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

q[j] = j + 1;

int p = rank - 1;

int count = 1;

while (p >= 0) {

int k = 0;

for (int l = 0, s = q[l] - 1; l < rank; l++, s = q[l] - 1)

for (int t = k; t < M; t++) {

if (a[t][s].Numerator != 0) {

swap(a[t], a[k]);

for (int i = M; i >= 0; i--) {

for (int j = N - 1; j >= 0; j--) {

if (i != k && j != s)

a[i][j] = a[i][j] - a[k][j] \* a[i][s] / a[k][s];

}

}

Rational tmp = a[k][s];

for (int i = N - 1; i >= 0; i--) {

a[k][i] /= tmp;

}

for (int i = M; i >= 0; i--) {

if (i != k) a[i][s] = 0;

}

k++;

break;

}

}

}

if (k == rank) {

int tmp = 1;

bool f = 1;

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

if (a[i][N - 1] < 0) f = 0; // Берём только опорные решения

}

if (f) {

vector<Rational> b;

cout << "X" << count << " = |";

for (int l = 0, s = q[l]; l < rank; l++, s = q[l]) {

for (int j = tmp; j < s; j++) {

cout << 0 << " | ";

}

tmp = s + 1;

b.push\_back(q[l]);

cout << a[l][N - 1] << " | ";

}

for (int j = tmp; j <= N - 1; j++) {

cout << 0 << " | ";

}

cout << endl;

b.push\_back(0);

Rational\*\* sys = new Rational \* [MS];

sys[0] = new Rational[MS \* (N + 1)];

for (int i = 1; i < MS; i++) {

sys[i] = sys[i - 1] + (N + 1);

}

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

for (int j = 2; j < N + 1; j++) {

sys[i][j] = a[i][j - 2];

}

}

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

sys[i][1] = a[i][N - 1];

}

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

sys[i][0] = b[i];

}

for (int i = 0; i < N + 1; i++) {

sys[MS - 1][i] \*= -1;

}

bases.push\_back(sys);

count++;

}

}

if (q[rank - 1] != N - 1)

p = rank - 1, q[p]++;

else {

q[--p]++;

for (int i = p + 1; i < rank; i++) q[i] = q[i - 1] + 1;

}

}

cout << "Найдено " << count - 1 << " опорных решений." << endl;

cout << "Выберете решение [1-" << bases.size() << "].\n";

int change = -1;

cin >> change;

if (change<1 || change>bases.size())return;

//output(bases[nob-1], MS, N+1);

Simplex(bases[change - 1], MS, N + 1);

}

void solve(Rational\*\* a, int MS, int N) {

int M = MS - 1;

Rational\*\* sys = new Rational \* [MS];

sys[0] = new Rational[MS \* N];

for (int i = 1; i < MS; i++) {

sys[i] = sys[i - 1] + N;

}

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

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

sys[i][j] = a[i][j];

}

}

int k = 0;

for (int s = 0; s < N - 1; s++) {

for (int t = k; t < M; t++) {

if (sys[t][s].Numerator != 0) {

swap(sys[t], sys[k]);

for (int i = M; i >= 0; i--) {

for (int j = N - 1; j >= 0; j--) {

if (i != k)

sys[i][j] = sys[i][j] - sys[k][j] \* sys[i][s] / sys[k][s];

}

}

Rational& tmp = sys[k][s];

for (int i = N - 1; i >= 0; i--) {

sys[k][i] = sys[k][i] / tmp;

}

k++;

output(sys, MS, N);

break;

}

}

}

for (int i = k; i < M; i++) {

if (sys[i][N - 1].Numerator != 0) {

cout << "Нет решений" << endl;

return;

}

}

int t = 0;

bool vars = false;

for (int i = 0; i < N - 1; i++) {

vars = false;

cout << "x" << i + 1 << " = ";

if (t < M && sys[t][i].Numerator) {

for (int j = i + 1; j < N - 1; j++) {

if (sys[t][j].Numerator != 0) {

outVar(-sys[t][j], vars);

cout << "x" << j + 1;

vars = true;

}

}

if (sys[t][N - 1].Numerator != 0) {

outVar(sys[t][N - 1], vars);

}

else if (!vars) {

cout << 0;

}

t++;

}

else {

cout << "любое число";

}

cout << endl;

}

cout << endl;

if (k && k < N - 1) {

basis(sys, MS, N, k);

}

delete[] sys;

return;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int M, N, tmp;

cout << "Введите количество уравнений: ";

cin >> M;

cout << "Введите количесвто переменных: ";

cin >> N;

cout << "Введите матрицу: \n";

if (M <= 0 || N <= 0) return -1;

N++;

Rational\*\* sys = new Rational \* [M + 1];

sys[0] = new Rational[(M + 1) \* N];

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

cin >> tmp;

sys[0][j] = Rational(tmp, 1);

}

for (int i = 1; i < M; i++) {

sys[i] = sys[i - 1] + N;

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

cin >> tmp;

sys[i][j] = Rational(tmp, 1);

}

}

sys[M] = sys[M - 1] + N;

cout << "Введите функцию: \n";

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

cin >> tmp;

sys[M][i] = tmp;

}

output(sys, M + 1, N);

solve(sys, M + 1, N);

delete[] sys;

return 0;

}