**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №9

«Вычисление обратной матрицы методом Гаусса-Жордана»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-15Б |  | преподаватель каф. ИУ5 |
| Коновалов Илья |  | Папшев И.С. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

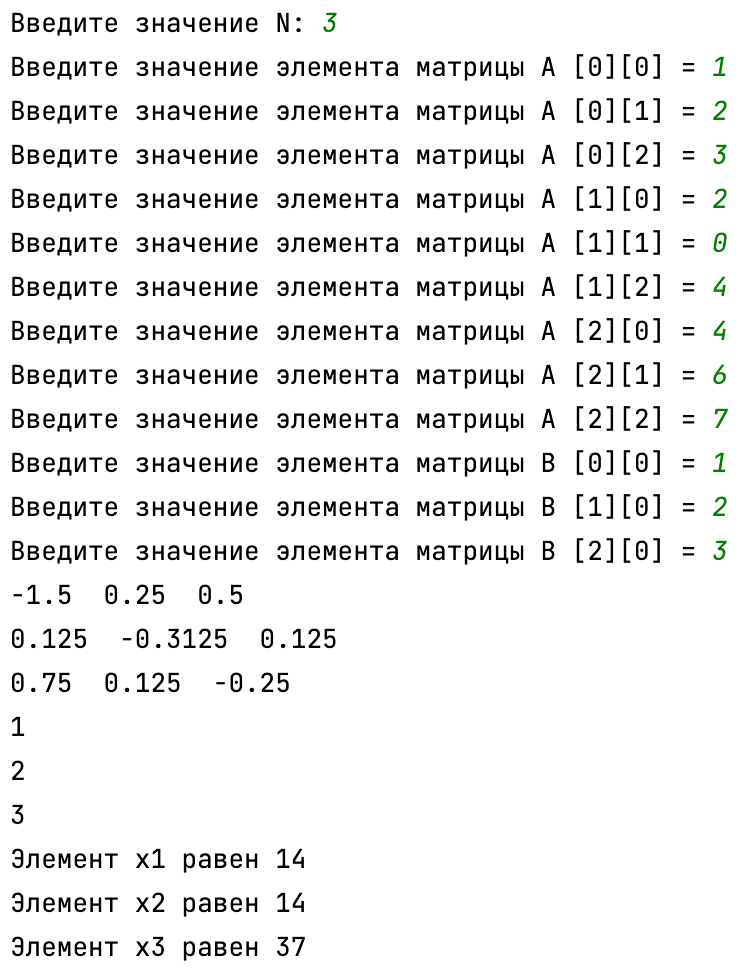
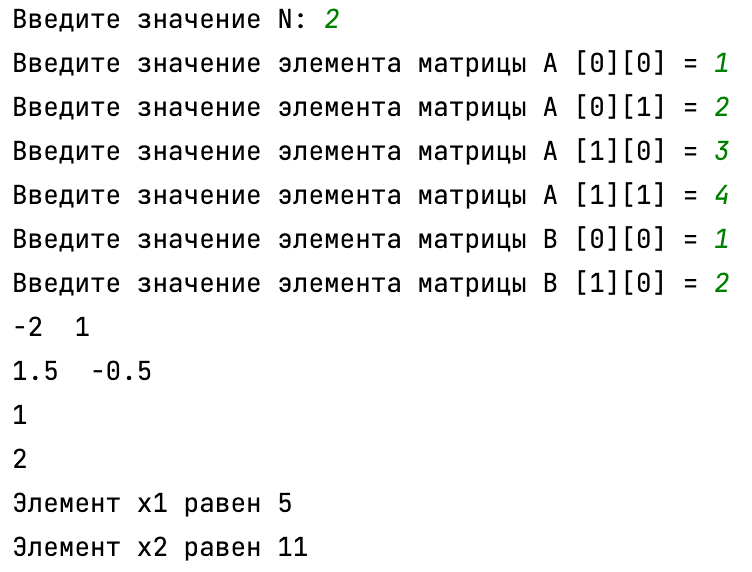
**Задача:**

Создать функцию для вычисления обратной матрицы по методу Гаусса-Жордана. Размер матрицы передавать в функцию в качестве параметра. Алгоритм решения системы линейных алгебраических уравнений оформить в виде функции. Выполнить тестирование программы. Проверку правильности результатов вычислений выполнять путем умножения полученной обратной матрицы на исходную (в результате должна получиться единичная матрица)Доработать алгоритм для случая, когда на главной диагонали появляется ноль. Для этого в прямом ходе перед делением выполнить проверку на ноль элемента на пересечении столбца коэффициентов исключаемой переменной и главной диагонали исходной матрицы. Применить функцию вычисления обратной матрицы для решения системы линейных алгебраических уравнений.

**Код программы:**

#include **<iostream>  
using namespace** std;  
  
**double**\*\* reverse(**double** \*\*A, **int** N){  
 **double** curr;  
 **double** \*\*E = **new double** \*[N];  
  
 **for** (**int** i = 0; i < N; ++i) {  
 E[i] = **new double**[N];  
 **for** (**int** j = 0; j < N; ++j) {  
 E[i][j] = (i == j);  
 }  
 }  
  
 **for** (**int** row = 0; row < N; ++row) {  
 **double** starting = A[row][row];  
 **if** (!starting) {  
 **bool** isFound = 0;  
 **for** (**int** i = row + 1; i < N; ++i) {  
 **if** (A[i][row]) {  
 **for** (**int** j = 0; j < N; ++j) {  
 swap(A[i][j], A[row][j]);  
 swap(E[i][j], E[row][j]);  
 }  
 isFound = **true**;  
 **break**;  
 }  
 }  
 **if** (!isFound) {  
 cout << **"Обратная матрица не существует!"** << endl;  
 **return nullptr**;  
 }  
 starting = A[row][row];  
 }  
 **for** (**int** i = row; i < N; ++i) {  
 A[row][i] /= starting;  
 }  
 **for** (**int** i = 0; i < N; ++i)  
 E[row][i] /= starting;  
  
 **for** (**int** i = row + 1; i < N; ++i) {  
 **const double** k = A[i][row];  
 **for** (**int** j = 0; j < N; ++j) {  
 A[i][j] -= A[row][j] \* k;  
 E[i][j] -= E[row][j] \* k;  
 }  
 }  
 }  
  
 **for** (**int** row = 1; row < N; ++row) {  
 **for** (**int** i = 0; i < row; ++i) {  
 **const double** k = A[i][row];  
 **for** (**int** j = 0; j < N; ++j) {  
 A[i][j] -= A[row][j] \* k;  
 E[i][j] -= E[row][j] \* k;  
 }  
 }  
 }  
  
 **return** E;  
}  
  
**double**\*\* prodMatr(**double**\*\* a, **double**\*\* b, **int** n1, **int** n2) {  
 **double**\*\* res;  
 **int** i, j;  
 res = **new double**\* [n1];  
 **for** (i = 0; i < n2; i++) {  
 res[i] = **new double**[n2];  
 }  
  
 **for** (i = 0; i < n1; i++) {  
 **for** (j = 0; j < n2; j++) {  
 res[i][j] = 0;  
 **for** (**int** k = 0; k < n1; k++)  
 res[i][j] += a[i][k] \* b[k][j];  
 }  
 }  
  
 **return** res;  
}  
  
**int** main(){  
 **int** N;  
 cout << **"Введите значение N: "**;  
 cin >> N;  
 **double** \*\*matrixA = **new double** \*[N];  
 **double** \*\*matrixB = **new double** \*[N];  
 **double** \*\*matrixC = **new double** \*[N];  
  
 **for** (**int** i = 0; i < N; i++)  
 matrixA[i] = **new double** [N];  
 **for** (**int** i = 0; i < N; i++)  
 matrixB[i] = **new double** [N];  
 **for** (**int** i = 0; i < N; i++)  
 **for** (**int** j = 0; j < N; j++){  
 cout << **"Введите значение элемента матрицы A ["** << i << **"]["** << j << **"] = "**;  
 cin >> matrixA[i][j];  
 }  
  
 **for** (**int** i = 0; i < N; i++)  
 **for** (**int** j = 0; j < 1; j++){  
 cout << **"Введите значение элемента матрицы B ["** << i << **"][0] = "**;  
 cin >> matrixB[i][j];  
 }  
 matrixA = reverse(matrixA, N);  
 **for** (**int** i = 0; i < N; i++) {  
 **for** (**int** j = 0; j < N; j++)  
 cout << matrixA[i][j] << **" "**;  
  
 cout << endl;  
 }  
 **for** (**int** i = 0; i < N; i++) {  
 **for** (**int** j = 0; j < 1; j++)  
 cout << matrixB[i][j] << **" "**;  
  
 cout << endl;  
 }  
 matrixA = reverse(matrixA, N);  
 matrixC = prodMatr(matrixA, matrixB, N, N);  
 **for** (**int** i = 0; i < N; i++) {  
 **for** (**int** j = 0; j < 1; j++) {  
 cout << **"Элемент x"** << i + 1 << **" равен "** << matrixC[i][j] << **"\t"**;  
 }  
 cout << endl;  
 }  
  
 **for** (**int** i = 0; i < N; i++)  
 **delete**[] matrixB[i];  
 **delete**[] matrixB;  
 cin.get();  
 **for** (**int** i = 0; i < N; i++)  
 **delete**[] matrixA[i];  
 **delete**[] matrixA;  
 cin.get();  
 **return** 0;  
}

**Результаты:**

**Вывод:**

В ходе выполнения лабораторной работы я приобрел навыки разработки функции, реализующей алгоритм вычисления обратной матрицы методом Гаусса-Жордана, разработки функции умножения матриц и использования разработанных функций для решения систем линейных уравнений.