**Отчет курсанта Громова Григорий Андреевича группы 22.Б05 о выполнении практического задания на 20.10.2022**

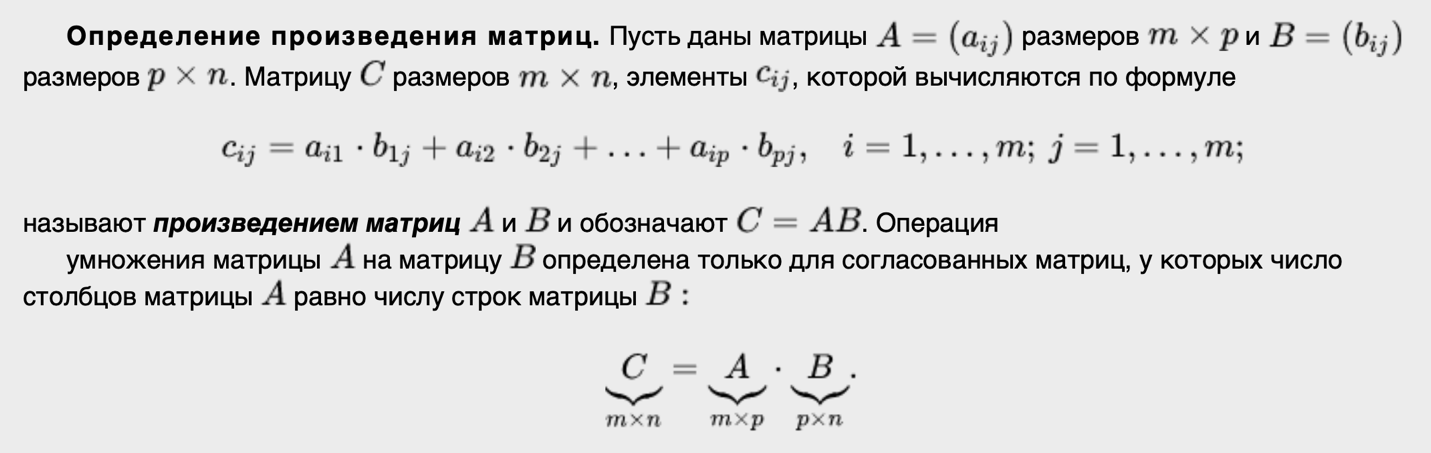
**Задания на массивы и функции**

Я не сумел оперативно выяснить, какие задания уже выполнены, а какие свободны, поэтому выбрал произвольные, но, честное слово, я написал всё сам.

Действия над матрицами

1. Умножение матриц

Математическая модель:



Алгоритм (тут изложены лишь основные шаги, более подробный алгоритм изложен в комментариях к тесту программы):

1. Создаем и инициализируем (посредством ввода с клавиатуры) переменные, отвечающие за размеры вводимых матриц
2. Проверяем соответствие введенных данных математической модели. Если они не соответствуют, выходим из программы
3. Создаем и инициализируем (посредством ввода с клавиатуры) массивы под матрицы, размеры которых берем из пункта 1 (размеры вводимых матриц)
4. Создаем новый массив P размером *число строк матрицы 1* на *число столбцов матрицы 2* (согласно математической модели)
5. Инициализируем каждую его ячейку согласно математической модели
6. Выводим массив P в виде таблицы

#include <stdio.h>

#include <stdlib.h>

int main(){

int m1; //число строк матрицы 1

int n1; //число столбцов матрицы 1

int m2; //число строк матрицы 2

int n2; //число столбцов матрицы 2

//инициализируем переменные:

printf("Введите количество строк в первой матрице\n");

scanf("%d", &m1);

printf("Введите количество столбцов в первой матрице\n");

scanf("%d", &n1);

printf("Введите количество строк во второй матрице\n");

scanf("%d", &m2);

printf("Введите количество столбцов во второй матрице\n");

scanf("%d", &n2);

if (n1 != m2){ //если нарушается условие умножения двух матриц, завершаем программу, предупредив пользователя

printf("Операция умножения двух матриц выполнима только в том случае, если число столбцов в первом сомножителе равно числу строк во втором");

return 0;

}

//создадим массивы под эти матрицы

int A1[m1][n1];

int A2[m2][n2];

int a, b; //вспомогательные счетчики для циклов

//заполним массив 1

for (a = 0; a < m1; a ++){

printf("Введите элементы строки %d матрицы 1\n", a + 1);

for (b = 0; b < n1; b++){

scanf("%d", &A1[a][b]); //считываем введенное пользователем число и записываем

//его в соответствующую порядку ввода ячейку массива A1

}

}

//заполним массив 2

for (a = 0; a < m2; a++){

printf("Введите элементы строки %d матрицы 2\n", a + 1);

for (b = 0; b < n2; b++){

scanf("%d", &A2[a][b]); //считываем введенное пользователем число и записываем

//его в соответствующую порядку ввода ячейку массива A2

}

}

printf("\n");

//выведем введенные матрицы

printf("Матрица 1:\n");

for (a = 0; a < m1; a++){

for (b = 0; b < n1; b++){

printf("%d ", A1[a][b]);

}

printf("\n");

}

printf("Матрица 2:\n");

for (a = 0; a < m2; a++){

for (b = 0; b < n2; b++){

printf("%d ", A2[a][b]);

}

printf("\n");

}

//создадим массив под матрицу произведения

int P[m1][n2];

int t = 0; //переменная для суммирования

int c = 0; //счетчик для того, чтобы пользоваться формулой

//вычисляем все элементы матрицы произведения, согласно математической модели

for (a = 0; a < m1; a++){

for (b = 0; b < n2; b++){

while(c < m2){

t += A1[a][c] \* A2[c][b];

c++;

}

P[a][b] = t;

t = 0;

c = 0;

}

}

//выводим результат

printf("Матрица произведения введенных вами матриц:\n");

for (a = 0; a < m1; a++){

for (b = 0; b < n2; b++){

printf("%d \t", P[a][b]);

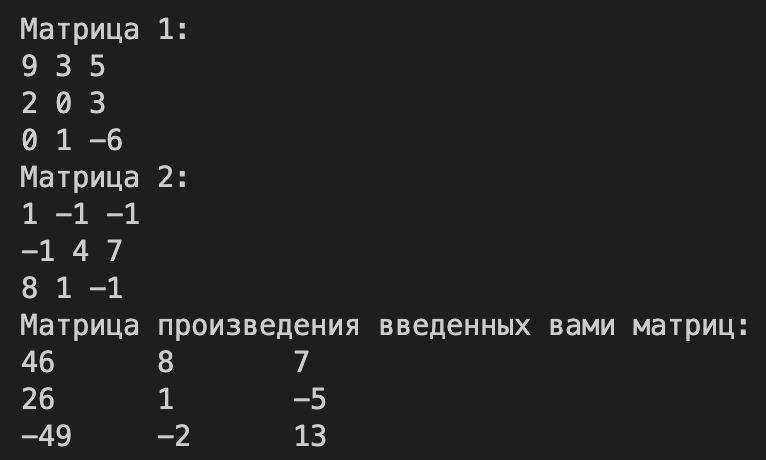
}

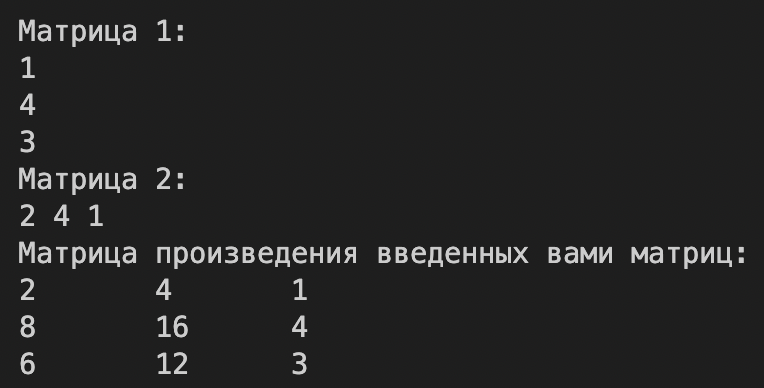
printf("\n");

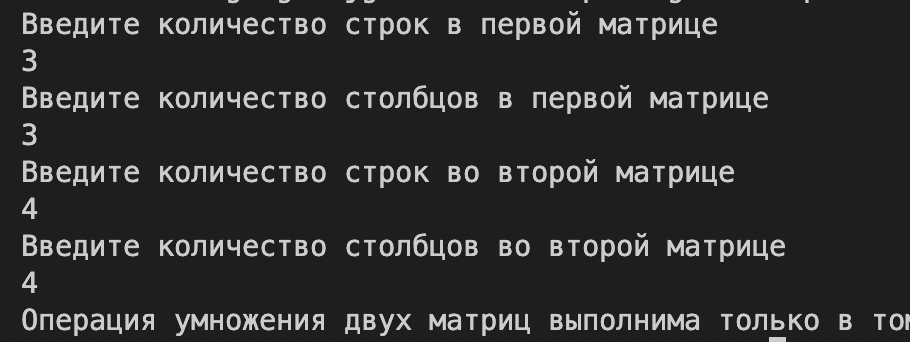
}

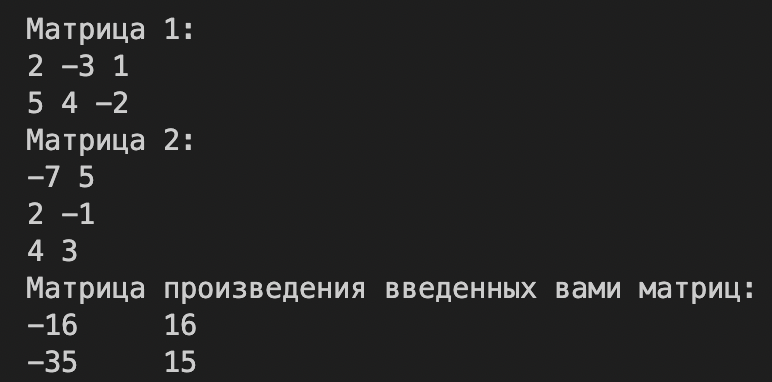
}

Скриншоты результатов запуска программы:



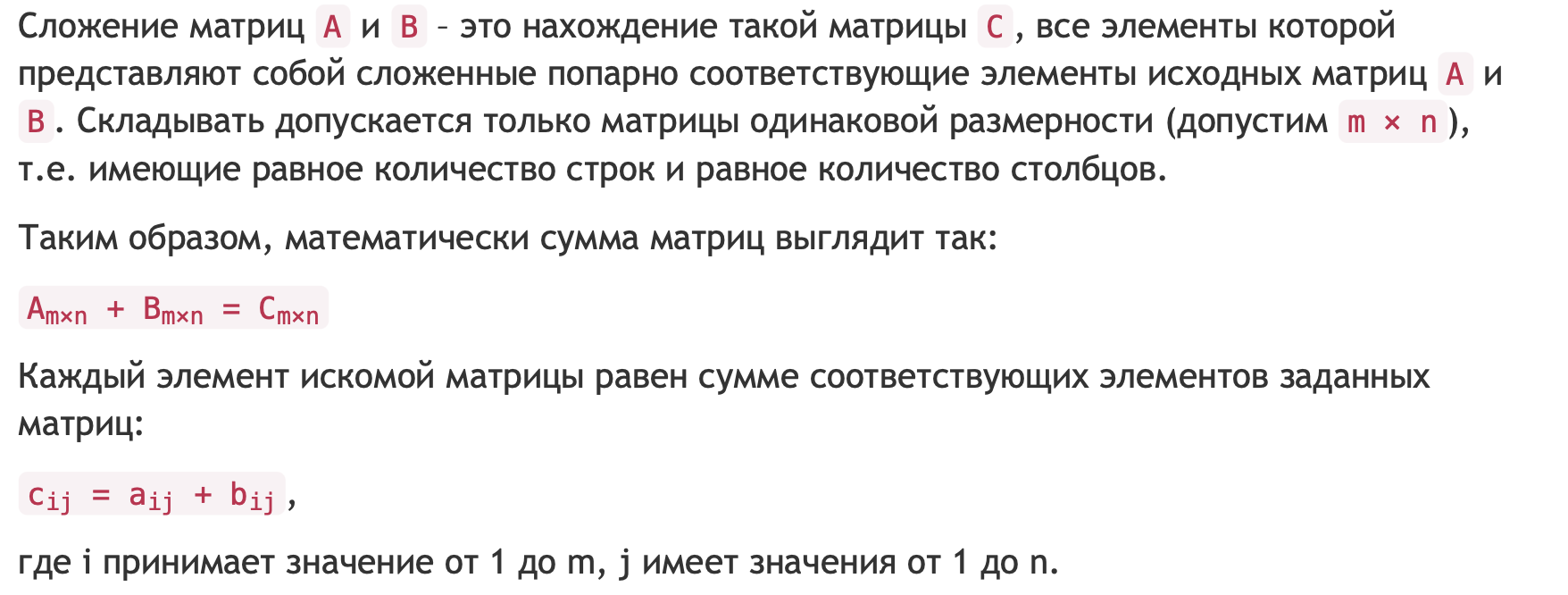






1. Сложение матриц

Математическая модель:



Алгоритм (тут изложены лишь основные шаги, более подробный алгоритм изложен в комментариях к тесту программы):

1. Создаем и инициализируем (посредством ввода с клавиатуры) переменные, отвечающие за размеры вводимых матриц
2. Создаем и инициализируем (посредством ввода с клавиатуры) массивы под матрицы, размеры которых берем из пункта 1 (размеры вводимых матриц)
3. Создаем новый массив P согласно математической модели
4. Инициализируем каждую его ячейку согласно математической модели
5. Выводим массив P в виде таблицы

Текст программы:

#include <stdio.h>

#include <stdlib.h>

int main(){

printf("Предупреждение: складывать допускается только матрицы одинаковой размерности \n");

int m; //число строк матрицы

int n; //число столбцов матрицы

//инициализируем переменные:

printf("Введите количество строк в матрицах, которые хотите сложить\n");

scanf("%d", &m);

printf("Введите количество столбцов в матрицах, которые хотите сложить\n");

scanf("%d", &n);

//создадим массивы под эти матрицы

int A1[m][n];

int A2[m][n];

int a, b; //вспомогательные счетчики для циклов

//заполним массив 1

for (a = 0; a < m; a ++){

printf("Введите элементы строки %d матрицы 1\n", a + 1);

for (b = 0; b < n; b++){

scanf("%d", &A1[a][b]); //считываем введенное пользователем число и записываем

//его в соответствующую порядку ввода ячейку массива A1

}

}

//заполним массив 2

for (a = 0; a < m; a++){

printf("Введите элементы строки %d матрицы 2\n", a + 1);

for (b = 0; b < n; b++){

scanf("%d", &A2[a][b]); //считываем введенное пользователем число и записываем

//его в соответствующую порядку ввода ячейку массива A2

}

}

printf("\n");

//выведем введенные матрицы

printf("Матрица 1:\n");

for (a = 0; a < m; a++){

for (b = 0; b < n; b++){

printf("%d ", A1[a][b]);

}

printf("\n");

}

printf("Матрица 2:\n");

for (a = 0; a < m; a++){

for (b = 0; b < n; b++){

printf("%d ", A2[a][b]);

}

printf("\n");

}

//создадим массив под матрицу суммы

int P[m][n];

//вычисляем все элементы матрицы суммы, согласно математической модели

for (a = 0; a < m; a++){

for (b = 0; b < n; b++){

P[a][b] = A1[a][b] + A2[a][b];

}

}

//выводим результат

printf("Матрица суммы введенных вами матриц:\n");

for (a = 0; a < m; a++){

for (b = 0; b < n; b++){

printf("%d \t", P[a][b]);

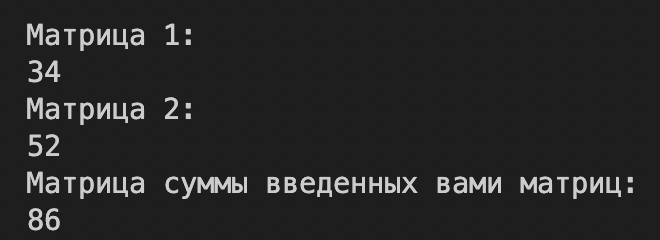
}

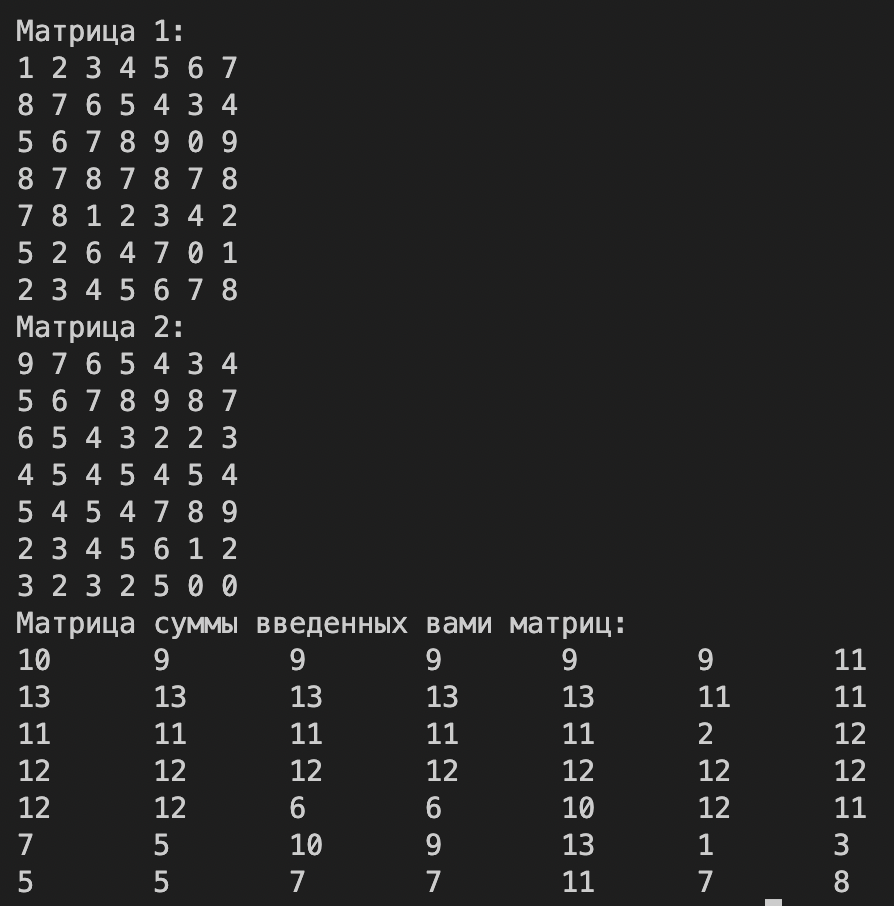
printf("\n");

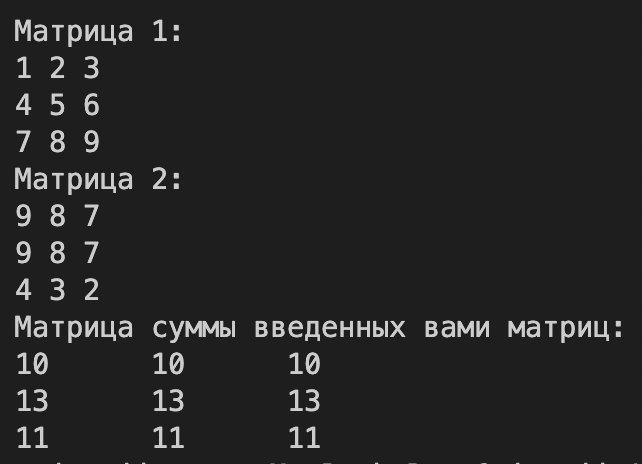
}

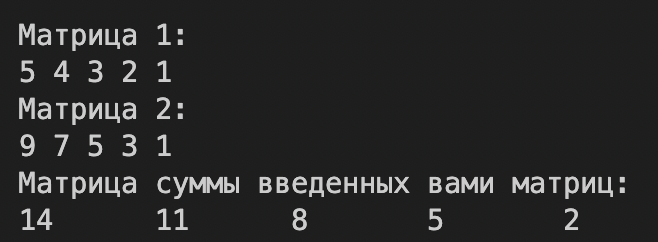
}

Скриншоты результатов запуска программы:



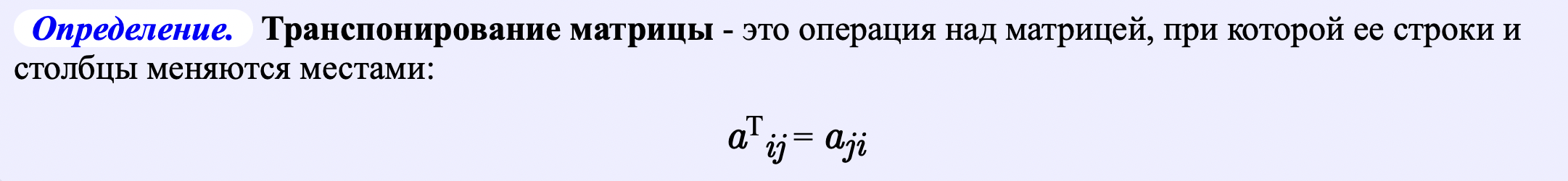






1. Транспонирование матрицы

Математическая модель:



Алгоритм (тут изложены лишь основные шаги, более подробный алгоритм изложен в комментариях к тесту программы):

1. Создаем и инициализируем (посредством ввода с клавиатуры) переменные, отвечающие за размеры вводимой матрицы
2. Создаем и инициализируем (посредством ввода с клавиатуры) массив под матрицу, размеры которого берем из пункта 1 (размеры вводимой матрицы)
3. Создаем новый массив T согласно математической модели
4. Инициализируем каждую его ячейку согласно математической модели
5. Выводим массив T в виде таблицы

Текст программы:

#include <stdio.h>

#include <stdlib.h>

int main(){

int m; //число строк матрицы

int n; //число столбцов матрицы

//инициализируем переменные:

printf("Введите количество строк в матрице, которую хотите транспонировать\n");

scanf("%d", &m);

printf("Введите количество столбцов в матрице, которую хотите транспонировать\n");

scanf("%d", &n);

//создадим массив под эти матрицу

int A[m][n];

int a, b; //вспомогательные счетчики для циклов

//заполним массив

for (a = 0; a < m; a ++){

printf("Введите элементы строки %d матрицы 1\n", a + 1);

for (b = 0; b < n; b++){

scanf("%d", &A[a][b]); //считываем введенное пользователем число и записываем

//его в соответствующую порядку ввода ячейку массива A1

}

}

printf("\n");

//выведем введенную матрицу

printf("Ваша матрица:\n");

for (a = 0; a < m; a++){

for (b = 0; b < n; b++){

printf("%d ", A[a][b]);

}

printf("\n");

}

//создадим массив под транспонированную матрицу

int T[n][m];

//вычисляем все элементы транспонированной матрицы по мат. модели

for (a = 0; a < n; a++){

for (b = 0; b < m; b++){

T[a][b] = A[b][a];

}

}

//выводим результат

printf("Транспонированная матрица:\n");

for (a = 0; a < n; a++){

for (b = 0; b < m; b++){

printf("%d \t", T[a][b]);

}

printf("\n");

}

}

Скриншоты результатов запуска программы:

