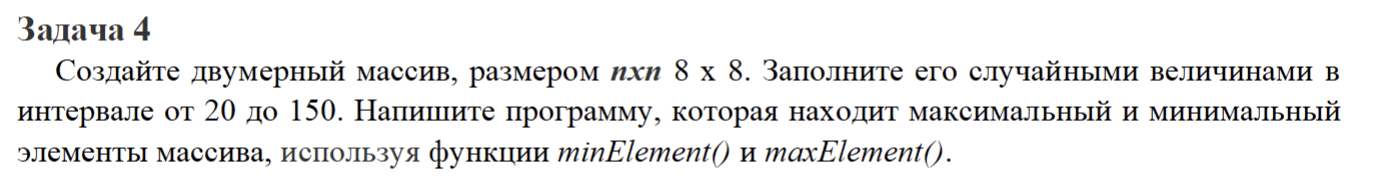
**Отчет курсанта Громова Григорий Андреевича группы 22.Б05 о выполнении практического задания на 27.10.2022**

***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.



**Алгоритм (более подробный алгоритм смотрите в комментариях к тексту программы):**

1. В main создаём масси 8 на 8 и заполняем случайными числами
2. Передаем его копию в display, где она выводится в виде таблицы на экран
3. Создаем в main переменные min и max
4. Инициализируем их, получая их значения из функций minElement и maxElement соответственно (предварительно передав в эти функции копию нашего массива)
5. Выводим переменные min и max на экран

**Математическая модель:** очевидна

**Текст программы:**

#include <stdio.h>

#include <stdlib.h>

void display(int num[8][8]); //объявляем функцию для вывода массива на экран

int minElement(int num[8][8]); //объявляем функцию поиска минимального элемента в массиве

int maxElement(int num[8][8]); //объявляем функцию поиска максимального элемента в массиве

int main (void)

{

int M[8][8]; //объявляем двумерный массив 8 на 8

int i, j; //объявляем переменные для циклов

int max, min; //объявляем переменные, куда запишем максимальный и минимальный элементы

for (i = 0; i < 8; i++){ //инициализируем массив, построчно заполняя его элементы случайными числами от 20 до 150

for (j=0; j < 8; j++){

M[i][j] = 20 + rand() % 131;

}

}

printf("Массив, наполненный случайными числами от 20 до 150:\n\n");

display(M); //выводим массив в виде таблице, обращаясь для этого к функции display

printf("\n");

//инициализируем переменные обращаясь к соответствующим функциям, передавая копию нашего массива

min = minElement(M);

max = maxElement(M);

//выводим значения этих переменных

printf("Минимальный элемент данного массива равен %d\n", min);

printf("Максимальный элемент данного массива равен %d\n", max);

return 0;

}

//инициализируем функции:

void display(int num[8][8]) //функция создает массив, дублирующий массив, переданный ей при вызове в main

{

int i, j; //объявляем переменные для циклов

for (i = 0; i < 8; i++){

for (j = 0; j < 8; j++){ //построчно выводим элементы массива

printf("%d\t", num[i][j]);

}

printf("\n");

}

}

int minElement(int num[8][8]){ //функция создает массив, дублирующий массив, переданный ей при вызове в main

int i, j; //объявляем переменные для циклов

int min\_el = 151; //создаем и инициализируем переменную, которая будет хранить минимальный элемент массива

for (i = 0; i < 8; i++){

for (j=0; j < 8; j++){

if (num[i][j] < min\_el){ //проходясь построчно по всем элементам массива, находим минимальный элемент

min\_el = num[i][j];

}

}

}

return min\_el; //возвращаем минимальный элемент

}

int maxElement(int num[8][8]){ //функция создает массив, дублирующий массив, переданный ей при вызове в main

int i, j; //объявляем переменные для циклов

int max\_el = 0; //создаем и инициализируем переменную, которая будет хранить максимальный элемент массива

for (i = 0; i < 8; i++){

for (j=0; j < 8; j++){

if (num[i][j] > max\_el){ //проходясь построчно по всем элементам массива, находим максимальный элемент

max\_el = num[i][j];

}

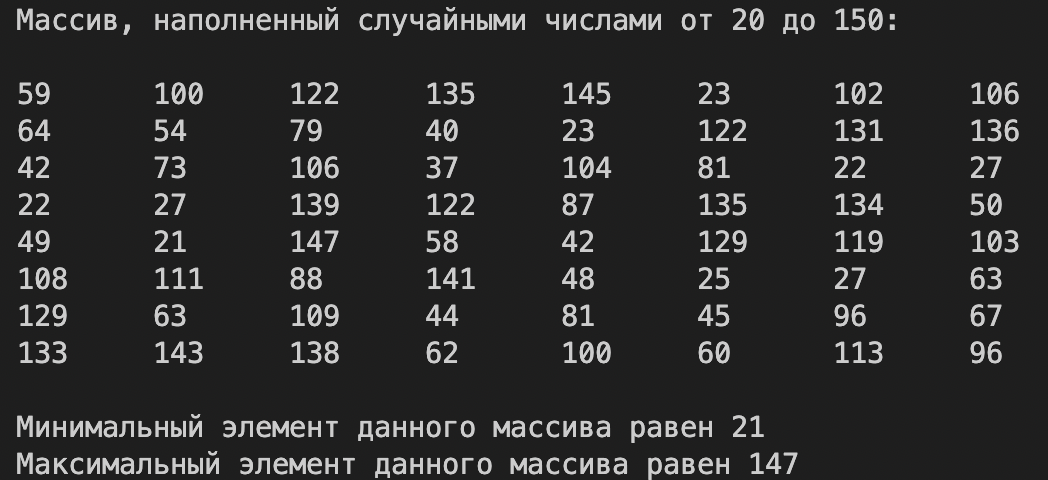
}

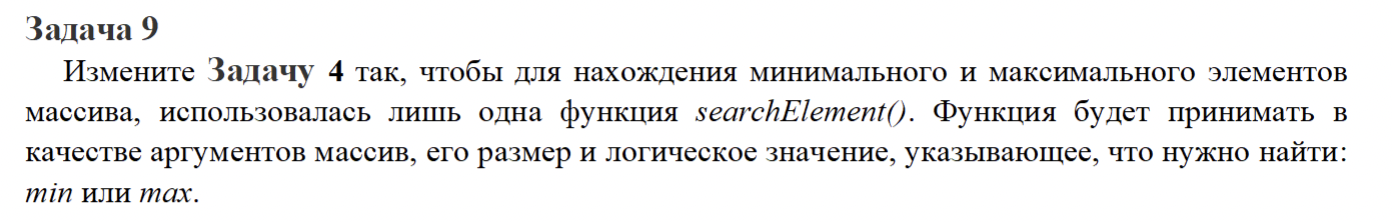
}

return max\_el; //возвращаем максимальный элемент

}

**Скриншоты результата работы программы** (из-за своеобразности работы функции rand при каждом новом запуске программы наполнение массива оставалось неизменным, поэтому пример только один):





Замечание: в моей реализации передача функции searchElement значения размера массива является избыточным, поэтому я убрал этот параметр

**Алгоритм (более подробный алгоритм смотрите в комментариях к тексту программы):**

1. В main создаём масси 8 на 8 и заполняем случайными числами
2. Передаем его копию в display, где она выводится в виде таблицы на экран
3. Просим пользователя ввести 0 (тогда далее выведем минимальный элемент данного массива) или 1 (тогда далее выведем максимальный элемент данного массива). Записываем введенное значение в переменную w. Если пользователь ввел не 0 и не 1, выводим сообщение об ошибке и завешаем программу.
4. Создаем в main переменную element
5. Инициализируем её, получая её значения из функций searchElement (предварительно передав в эту функцию копию нашего массива и число w)
6. searchElement в зависимости от значения w ищет либо минимальный, либо максимальный элемент массива и передает его в main
7. Выводим переменную element на экран

**Математическая модель:** очевидна

**Текст программы:**

#include <stdio.h>

#include <stdlib.h>

void display(int num[8][8]); //функция для вывода массива на экран

int searchElement(int num[8][8], int w); //функция поиска минимального элемента в массиве

int main (void)

{

int M[8][8]; //объявляем двумерный массив 8 на 8

int i, j; //объявляем переменные для циклов

int element; //объявляем переменные, куда запишем максимальный и минимальный элементы

for (i = 0; i < 8; i++){ //инициализируем массив, построчно заполняя его элементы случайными числами от 20 до 150

for (j=0; j < 8; j++){

M[i][j] = 20 + rand() % 131;

}

}

printf("Массив, наполненный случайными числами от 20 до 150:\n\n");

display(M); //выводим массив в виде таблице, обращаясь для этого к функции display

printf("\n");

int w;

printf("Если вы хотите найти минимальный элемент данного массива, введите 0, если максимальный, введите 1:\n");

scanf("%d", &w);

if ((w != 0)&&(w != 1)){

printf("Вы ввели некорректное число!\n");

return 0;

}

//инициализируем переменные обращаясь к соответствующим функциям, передавая копию нашего массива

element = searchElement(M, w);

printf("Искомое значение равно %d\n", element);

return 0;

}

void display(int num[8][8]) //функция создает массив, дублирующий массив, переданный ей при вызове в main

{

int i, j; //объявляем переменные для циклов

for (i = 0; i < 8; i++){

for (j = 0; j < 8; j++){ //построчно выводим элементы массива

printf("%d\t", num[i][j]);

}

printf("\n");

}

}

//найдем минимальный или максимальный элемент массива в соответствии со значением переменной w

int searchElement(int num[8][8], int w){ //функция создает массив, дублирующий массив, переданный ей при вызове в main

int i, j; //объявляем переменные для циклов

int el; //объявляем переменную, которая будет хранить искомый элемент массива

if (w == 0){ //если нужен минимальный элемент

el = 151; //инициализируем переменную, которая будет хранить искомый элемент массива

for (i = 0; i < 8; i++){

for (j=0; j < 8; j++){

if (num[i][j] < el){ //проходясь построчно по всем элементам массива, находим минимальный элемент

el = num[i][j];

}

}

}

return el; //возвращаем искомый элемент

}

else{ //если нужен максимальный элемент

el = 0; //инициализируем переменную, которая будет хранить искомый элемент массива

for (i = 0; i < 8; i++){

for (j=0; j < 8; j++){

if (num[i][j] > el){ //проходясь построчно по всем элементам массива, находим максимальный элемент

el = num[i][j];

}

}

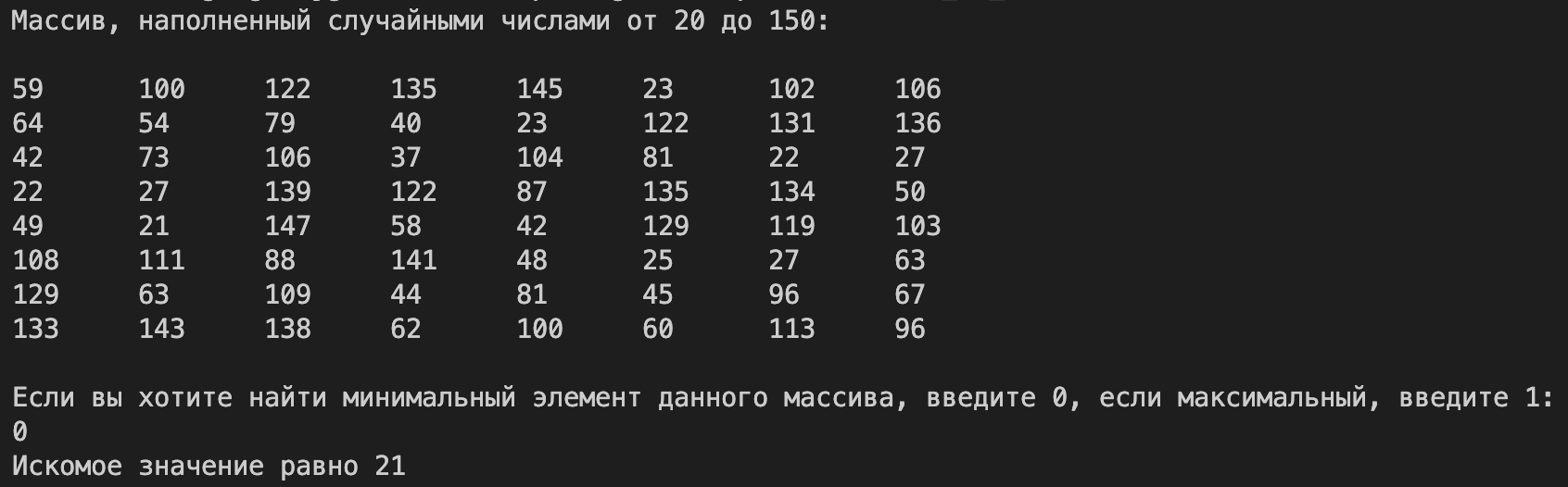
}

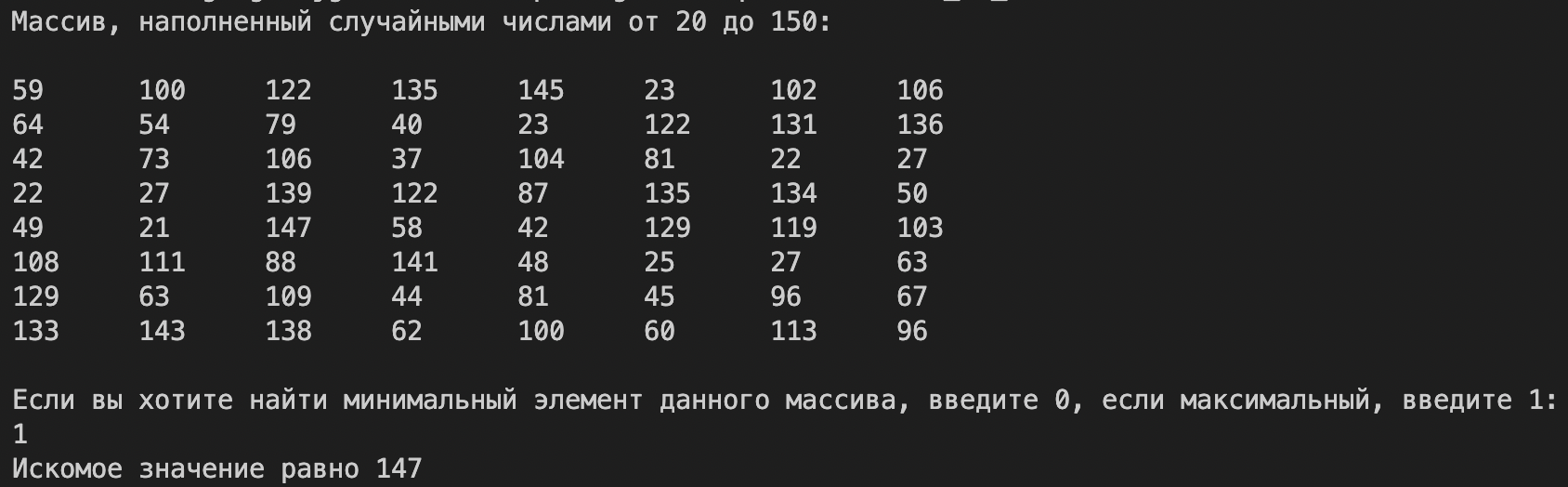
return el; //возвращаем искомый элемент

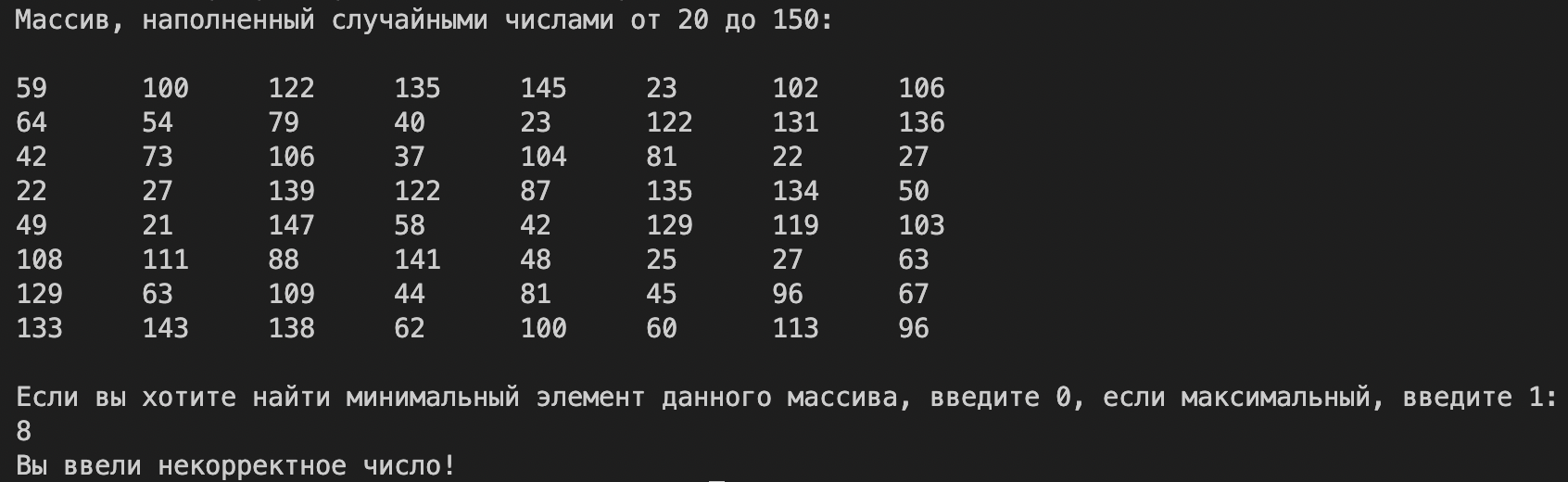
}

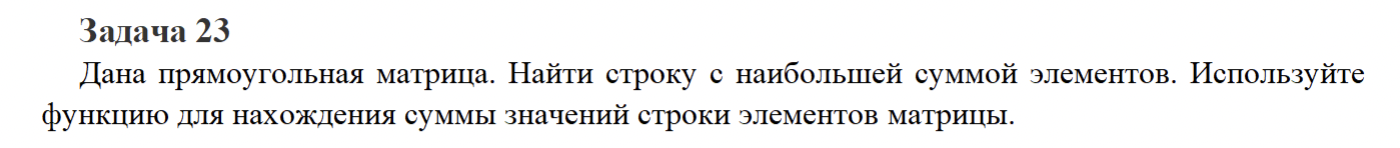
}

**Скриншоты результата работы программы:**









Алгоритм:

1. Заводим глобальные переменные m и n, которые будут отвечать за ширину и высоту матрицы соответственно
2. Объявляем вспомогательные функции
3. В main объявляем и инициализируем массив и выводим его на экран с помощью функции display
4. С помощью функции line\_sum вычисляем сумму элементов первой строки массива
5. В цикле (каждый раз отправляя в функцию line\_sum указатель на первый элемент i-ого вложенного массива) вычисляем суммы оставшихся строк и выбираем из них и первой наибольшую

Текст программы:

#include <stdio.h>

#include <stdlib.h>

int m = 5; //количество строк

int n = 5; //количество столбцов

void display(int num[m][n]); //объявляем функцию вывода на экран массива

int line\_sum(int \*uk, int size); //объявляем функцию поиска суммы элементов в строке

int main (){

int M[m][n]; //объявляем массив m строк на n столбцов

int a, b; //вспомогательные переменные для циклов

printf("Заполните построчно массив %d на %d\n", m, n);

//инициализируем наш массив с помощью ввода с клавиатуры

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

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

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

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

//его в соответствующую порядку ввода ячейку массива M

}

}

printf("Вот ваш массив:\n"); //выводим наш массив, обращаясь к функции display, передавая

//ей в качестве аргумента копию нашего массива

display(M);

int \*uk = &M[0][0]; //создаем указатель uk и записываем туда номер ячейки, где лежит

//первый элемент первого вложенного массива

int max\_sum = line\_sum(uk, n); //в max\_sum записываем значение, полученное от функции, которой

//мы передали указатель uk и длину строки n

int i; //вспомогательная переменная для цикла

int mid\_sum; //переменная, куда мы будем записывать сумму элементов каждой новой строки

int mem = 1; //переменная, где будет храниться номер строки с наибольшей суммой

for (i = 1; i < m; i++){

uk = &M[i][0]; //указатель на первый элемент i-ого вложенного массива (i-ой строки)

mid\_sum = line\_sum(uk, n); //сумма элементов i-ой строки

if (mid\_sum > max\_sum){ //сравним с наибольшей

max\_sum = mid\_sum;

mem = i + 1;

}

}

printf("Строка %d имеет максимальную сумму элементов, а именно %d\n", mem, max\_sum);

}

int line\_sum(int \*uk, int size){

int i; //вспомогательная переменная для цикла

int sum = 0; //переменная для суммы

for (i = 0; i < size; i++){ //с помощью указателя проходим по всем элементам строки

sum += \*(uk + i); //считаем их сумму

}

return sum; //возвращаем полученное значение

}

void display(int num[m][n]) //функция создает массив, дублирующий массив, переданный ей при вызове в main

//num - точная копия массива M, однако хранится в другом месте

{

int i, j; //объявляем переменные для циклов

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

for (j = 0; j < n; j++){ //построчно выводим элементы массива

printf("%d\t", num[i][j]);

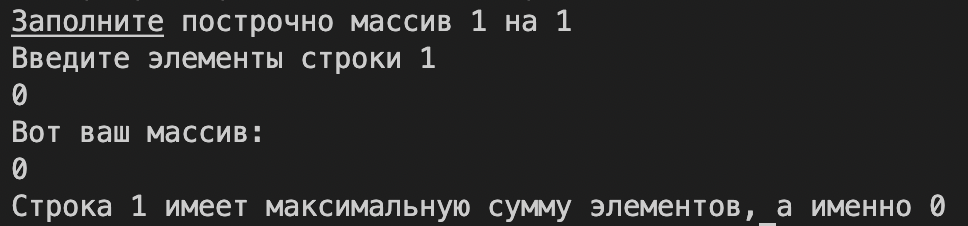
}

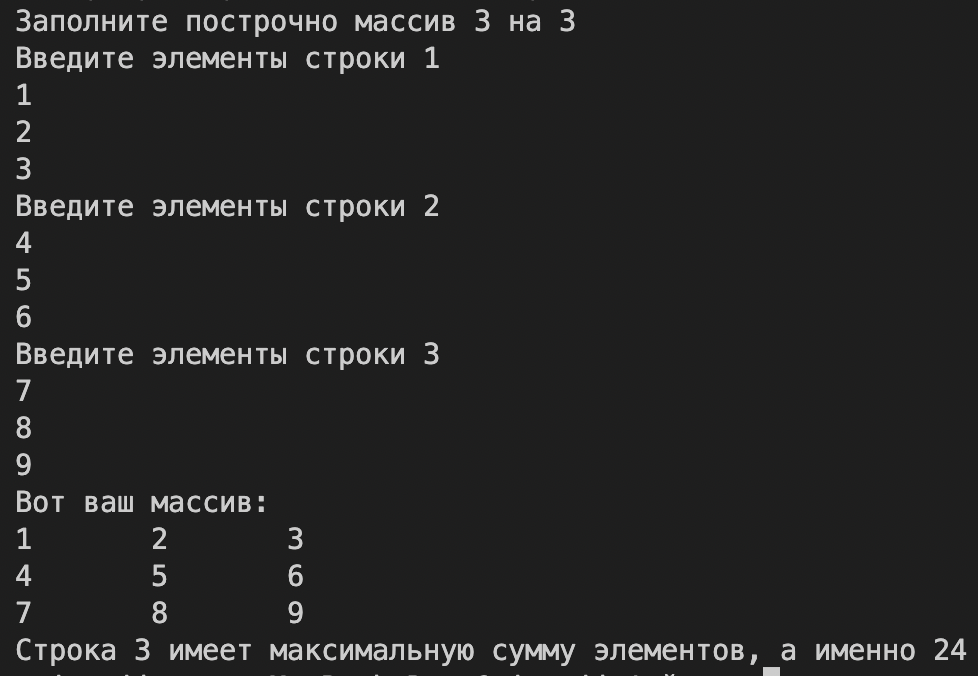
printf("\n");

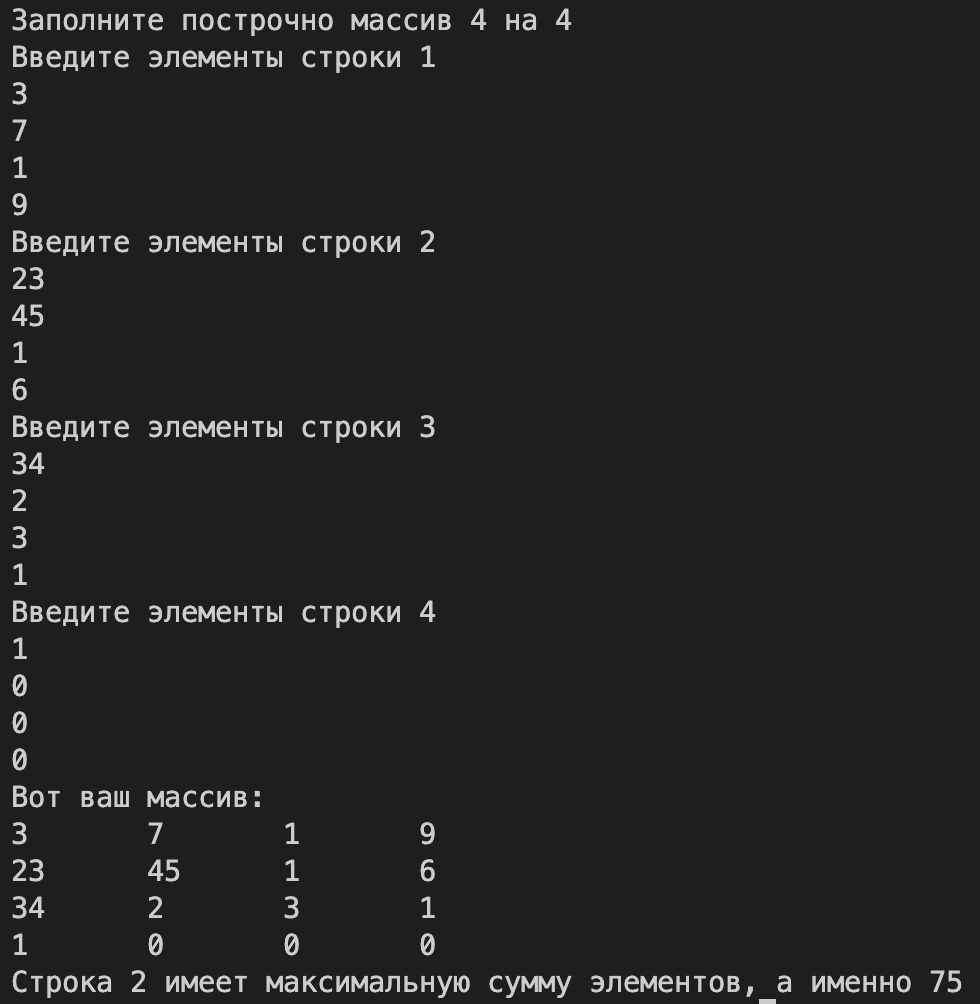
}

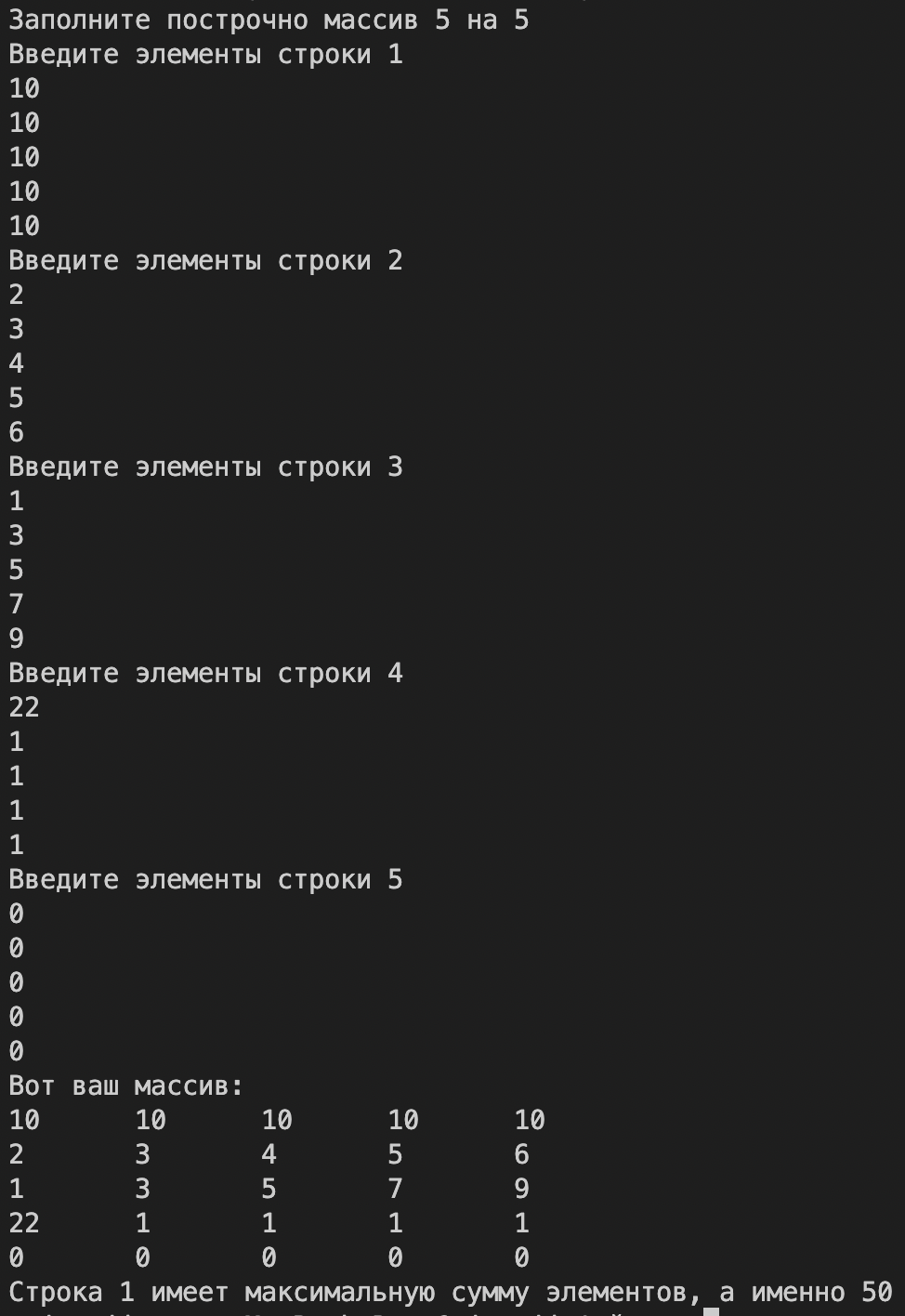
}

**Скриншоты результатов работы программы:**

****



****

****