Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра “Вычислительная Техника”

**ОТЧЕТ**

по лабораторной работе №12

по курсу «Программирование»

на тему «Многомодульные программы»

Выполнили студенты

группы 24ВВВ

Гурин А.Н.

Захаров А.В.

Принял:

к.т.н. доцент Юрова О.В.

Пенза 2024

**Цель работы:**

Изучение принципов написания многомодульных программ и приобретение навыков в написании таких программ и работе с ними.

**Лабораторное задание:**

1. Выбрать вариант задания в соответствии с номером бригады.

2. Каждая функция 8-ой лабораторной работы должна быть помещена в отдельный файл. Далее в среде необходимо создать проект, в который включаются все файлы с функциями.

3. Выполнить программу на компьютере и оценить правильность ее работы на различных наборах данных.

**ЛИСТИНГ**

**А)**

Общие библиотеки:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <Windows.h>

Create\_array.c:

int\* createArray(int size) {

int\* array = (int\*)malloc(size \* sizeof(int));

return array;

}

Fill\_array.c:

void fillArrayWithRandomNumbers(int\* array, int size) {

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

array[i] = rand() % 201 - 100;

}

}

Print\_array.c:

void printArray(int\* array, int size) {

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

printf("%d ", array[i]);

}

printf("\n");

}

Modify\_array.c:

void modifyArray(int\* array, int size, int k1, int k2) {

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

if (array[i] > 0) {

array[i] -= k1;

}

else {

array[i] -= k2;

}

}

}

Create\_array.h:

#ifndef Create\_array\_h

#define Create\_array\_h

int\* createArray(int size);

#endif

Print\_array.h:

#ifndef Print\_array\_h

#define Print\_array\_h

void printArray(int\* array, int size);

#endif

Fill\_array.h:

#ifndef Fill\_array\_h

#define Fill\_array\_h

void fillArrayWithRandomNumbers(int\* array, int size);

#endif

Modify\_array.h:

#ifndef Modify\_array\_h

#define Modify\_array\_h

void modifyArray(int\* array, int size, int k1, int k2);

#endif

MAIN.c

int main() {

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

srand(time(NULL));

const int size = 20;

int\* A = createArray(size);

fillArrayWithRandomNumbers(A, size);

printf("Исходный массив A:\n");

printArray(A, size);

int k1 = A[0];

int k2 = A[1];

modifyArray(A, size, k1, k2);

printf("Модифицированный массив A:\n");

printArray(A, size);

free(A);

return 0;

}

**Б)**

Общие библиотеки:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

Generate\_matrix.c:

void generateMatrix(int\* matrix, int rows, int cols) {

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

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

\*(matrix + i \* cols + j) = rand() % 21 - 10;

}

}

}

Print\_matrix.c:

void printMatrix(int\* matrix, int rows, int cols) {

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

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

printf("%3d ", \*(matrix + i \* cols + j));

}

printf("\n");

}

}

Find\_min\_column\_sum.c:

void findMinColumnSum(int\* matrix, int rows, int cols, int\* minSumIndex, int\* minsum) {

\*minsum = INT\_MAX;

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

int columnSum = 0;

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

columnSum += \*(matrix + i \* cols + j);

}

if (columnSum < \*minsum) {

\*minsum = columnSum;

\*minSumIndex = j + 1;

}

}

}

Generate\_matrix.h:

#ifndef Generate\_matrix\_h

#define Generate\_matrix\_h

void generateMatrix(int\* matrix, int rows, int cols);

#endif

Print\_matrix.h:

#ifndef Print\_matrix\_h

#define Print\_matrix\_h

void printMatrix(int\* matrix, int rows, int cols);

#endif

Find\_min\_column\_sum.h:

#ifndef Find\_min\_column\_sum\_h

#define Find\_min\_column\_sum\_h

void findMinColumnSum(int\* matrix, int rows, int cols, int\* minSumIndex, int\* minsum);

#endif

MAIN.c

int main() {

setlocale(LC\_ALL, "Russian");

srand((unsigned)time(NULL));

const int rows = 6;

const int cols = 5;

int matrix[rows][cols];

int\* pmatrix = (int\*)matrix;

generateMatrix(pmatrix, rows, colss);

printMatrix(pmatrix, rows, cols);

int minSumIndex = 0;

int minsum = INT\_MAX;

findMinColumnSum(pmatrix, rows, cols, &minSumIndex, &minsum);

printf("Столбец с минимальной суммой элементов:%d \n", minSumIndex);

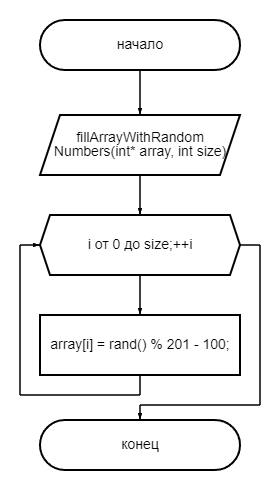
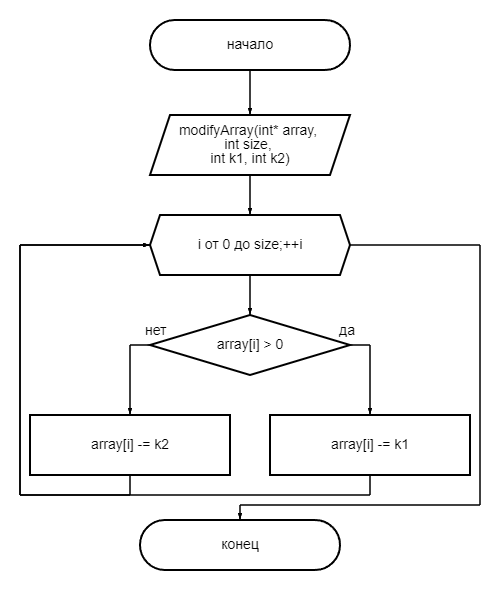
printf("Сумма этого столбца:%d\n", minsum);

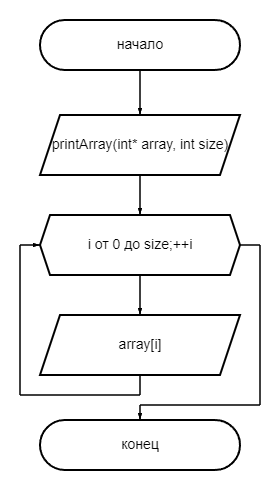
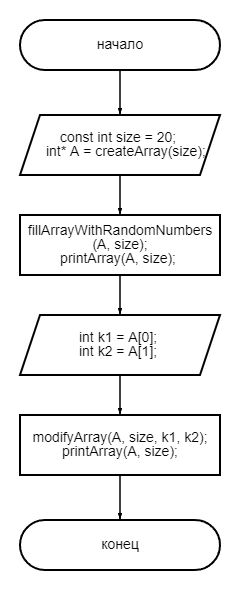
return 0;

}

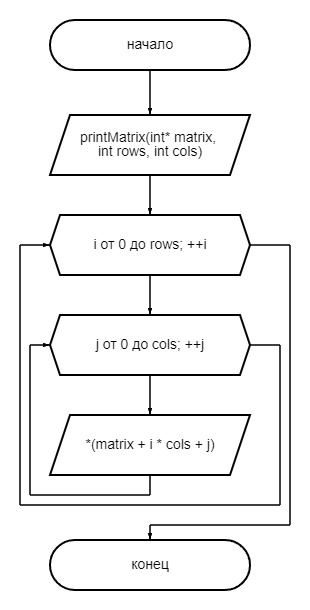
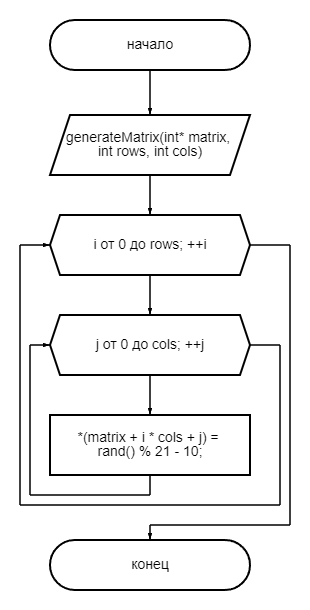
**СХЕМА**

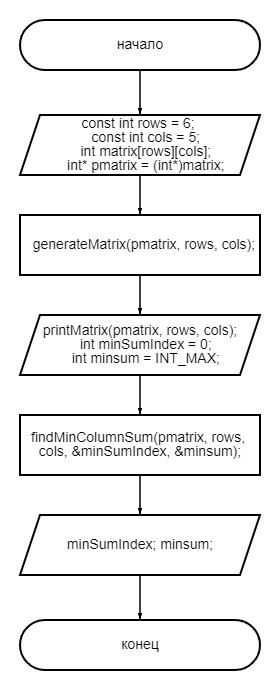
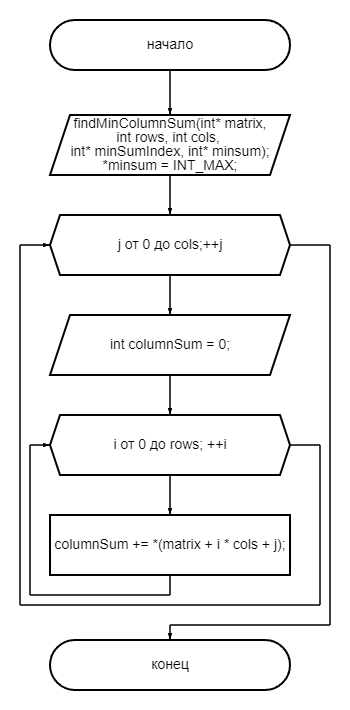
**А)**





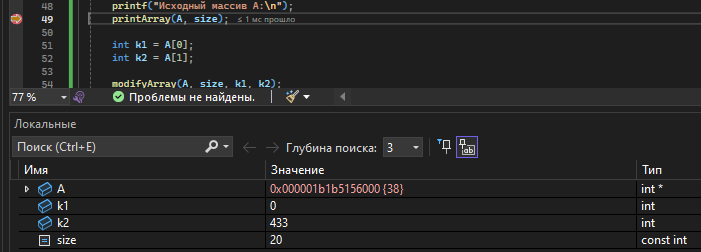
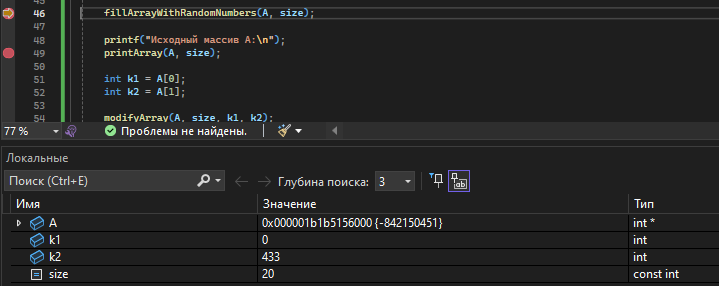
**Б)**

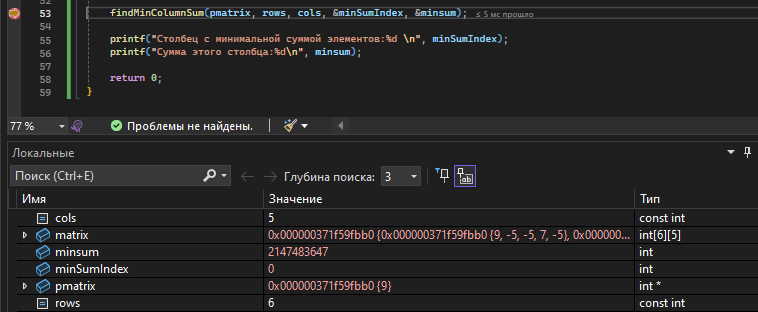


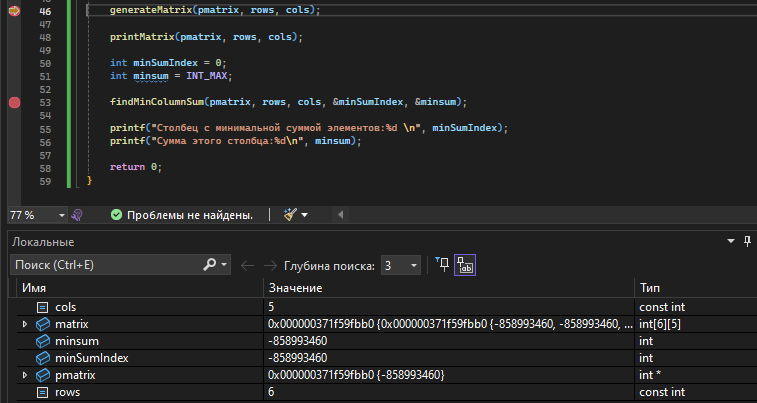


**ТРАССИРОВКА**

**А)**

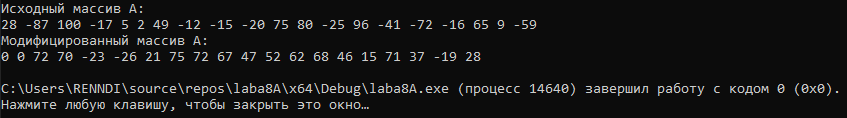


**Б)**

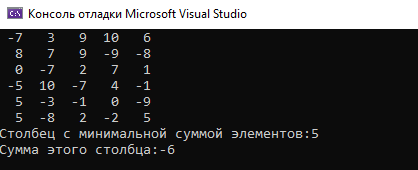


**РЕЗУЛЬТАТ**

**А)**



**Б)**



**ВЫВОД:**

В ходе выполнения лабораторной работы мы изучили принципы написания многомодульных программ и приобрели навыки в написании таких программ и работе с ними.