Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЁТ

к лабораторной работе №4

на тему

МНОГОМЕРНЫЕ МАССИВЫ

ВАРИАНТ 1

Проверил Выполнил

асс. каф. ЭВМ студент гр. 318302 П.С. Жуковец В.С.Бахур

МИНСК 2023

**1** **ЗАДАНИЕ №1**

**1.1 Условие**

Двумерный массив, содержащий равное число строк и столбцов, называется магическим квадратом, если суммы чисел, записанных в каждой строке, каждом столбце и каждой из двух больших диагоналей, равны одному и тому же числу. Определить, является ли данный массив А из n строк и n столбцов магическим квадратом.

**1.2 Алгоритм решения**

Алгоритм предоставлен в «Приложении A»; Рисунок 1.1

**1.3 Листинг программы**

/\* В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- количество элементов массива, равных 0;

- сумму элементов массива, расположенных после минимального элемента.\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <math.h>

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "Rus");

int elQuantity, choice, i, array[100], elEqualZero = 0, elMin;

printf("Количество элементов массива:");

scanf("%d", &elQuantity);

if (elQuantity <= 0) {

printf("Неверный ввод!");

return 1;}

printf("Если хотите инициализировать массив случайными значениями нажмите '1', в ином случае любой дугой символ:");

scanf("%d", &choice);

if (choice == 1) {

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

array[i] = -50 + rand() % 101;

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

printf("\n");}

}

else {

printf("Введите элементы массива:\n");

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

scanf("%d", &array[i]);}

}

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

if (array[i] == 0){

elEqualZero += 1;}

}

printf("Количество элементов равных нулю = %d\n", elEqualZero);

elMin = array[0]; //Минимальный элемент

int minIndex = 0; //Номер минимального числа

for (int i = 1; i < elQuantity; i++)

{

if (array[i] < elMin)

{

elMin = array[i];

minIndex = i; //Присвоение позиции

}

}

printf("Минимальный элемент = %d\n", elMin);

int sum = 0;

for (i = minIndex + 1; i < elQuantity; i++)

{

sum += array[i];

}

printf("Сумма элементов после минимального элемента: %d\n", sum);

return 0;

}

**2 ЗАДАНИЕ №2**

**2.1 Условие**

Найти в матрице первую строку, все элементы которой положительны, и сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.

**2.2 Алгоритм решения**

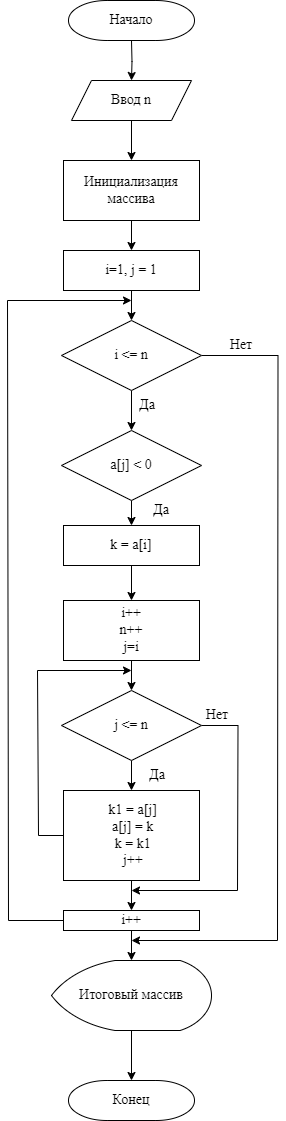


Рисунок 2.1 – Алгоритм задачи 2

**2.3 Листинг программы**

//В массиве из n элементов циклически сдвинуть все четные элементы на k влево. Нечетные оставить на своих местах (например, k=2: 0,1,2,3,4,5,6,7 =>4,1,6,3,0,5,2,7)

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <math.h>

void EvenElements(int arr[], int elQuantity, int k) {

k = k % elQuantity; // Убедимся, что k не больше кол-ва элементов

for (int i = 0; i < elQuantity; i++)

{

if (arr[i] % 2 == 0) // Проверяем, является ли элемент четным

{

int newIndex = (i - k + elQuantity) % elQuantity; // Вычисляем новый индекс после сдвига

int temp = arr[i];

arr[i] = arr[newIndex];

arr[newIndex] = temp;

}

}

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "Rus");

int elQuantity, i, arr[100], k;

int choice;

printf("Количество элементов массива (20 достаточно):");

scanf\_s("%d", &elQuantity);

printf("Если хотите инициализировать массив случайными значениями нажмите '1', в ином случае любой дугой символ:\n");

scanf\_s("%d", &choice);

if (choice == 1)

{

for (int i = 0; i < elQuantity; i++)

{

arr[i] = -50 + rand() % 100;

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

printf("\n");

}

}

else

{

printf("Введите элементы массива:\n");

for (i = 0; i < elQuantity; i++)

{

scanf\_s("%d", &arr[i]);

}

}

printf("Введите значение k: ");

scanf\_s("%d", &k);

printf("Исходный массив: ");

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

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

}

printf("\n");

EvenElements(arr, elQuantity, k);

printf("Массив после циклического сдвига четных элементов на k: ");

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

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

}

printf("\n");

return 0;

}

**3 ЗАДАНИЕ №3**

* 1. **Условие**

В квадратной матрице размером NxN найти сумму элементов в 1-ой области.

**3.2 Алгоритм решения**

Данная блок-схема характеризует алгоритм работы программы (рис. 3.1):

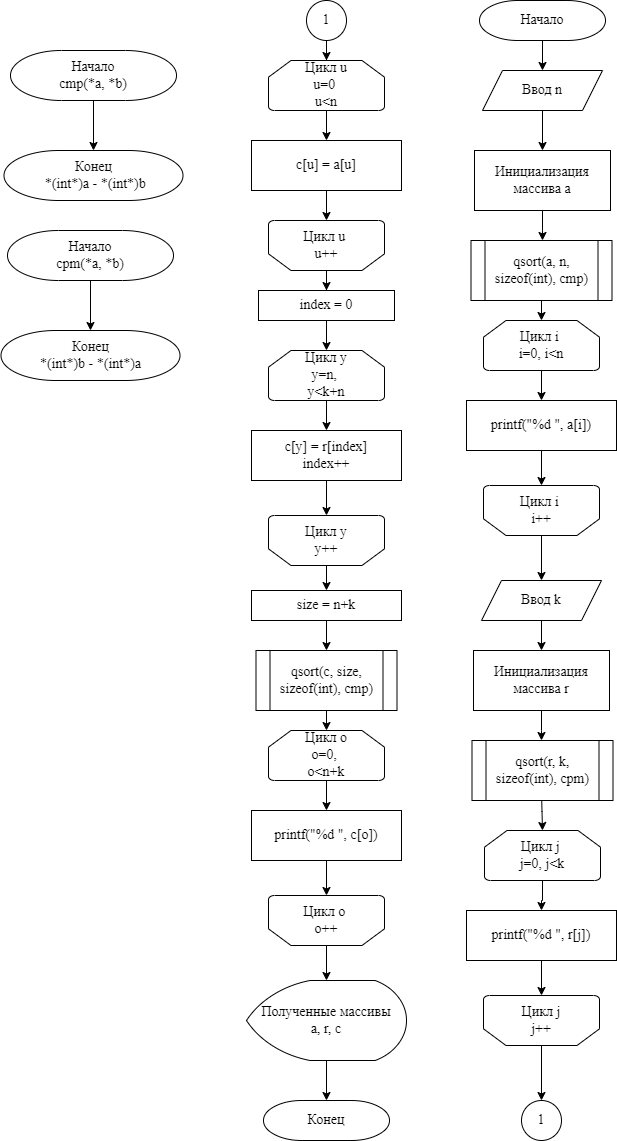


Рисунок 3.1 – Алгоритм задачи 3

**3.3 Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

#include <string.h>

#include <stdbool.h>

const int NumDigits = 30;

void readNum(char buffer[])

{

printf("Введите число\n");

int success = 1;

while(10)

{

scanf("%s[30]", buffer);

int strLength = strlen(buffer);

printf("strLength = %d\n", strLength);

if (strLength == 30)

{

for (int i = 0; buffer[i]; i++)

{

if ((buffer[i] < 47) || (57 < (buffer[i])))

{

printf("Ошибка, введён неверный символ, введите 30-ти значное число\n");

success = 0;

break;

}

buffer[i] -= '0';

}

if (success == 1)

{

return;

}

}

else

{

printf("Ошибка, неверная длинна, введите 30-ти значное число\n");

}

}

}

void subtract(char arr1[], char arr2[], char arrResult[] )

{

for (int i = NumDigits - 1; i >= 0; i--)

{

arrResult[i] = arr1[i] + arr2[i];

if (arrResult[i] > 10)

{

arrResult[i] -= 10;

arr1[i - 1] += 1;

}

}

}

int main()

{

system("chcp 1251 > nul");

char num1[31], num2[31], num3[31];

readNum(num1);

readNum(num2);

subtract(num1, num2, num3);

for (int i = 0; i < NumDigits; i++)

{

num3[i] += '0';

}

num3[NumDigits] = 0;

char\* ptr = num3;

while (\*ptr == '0' && \*(ptr + 1) != 0)

{

++ptr;

}

printf("%s", ptr);}

**Приложение А**

Блок-схема алгоритма 1

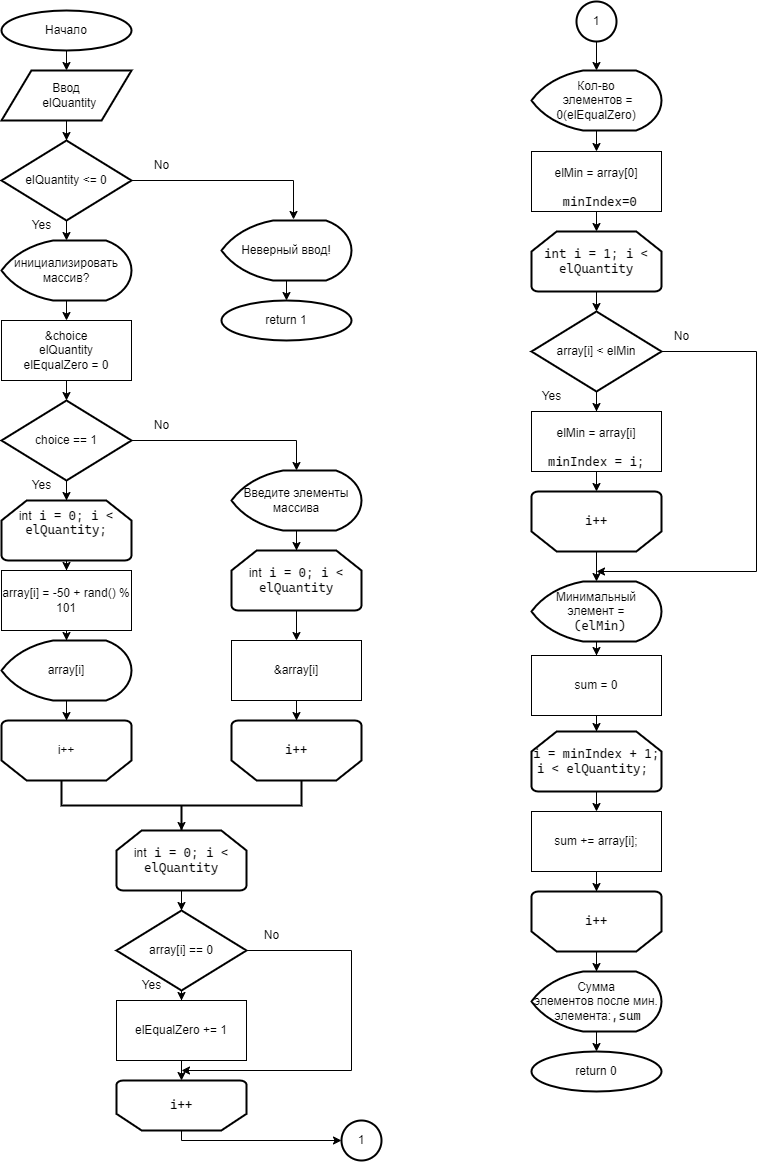


Рисунок 1.1 – Блок-схема №1