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 №3

на тему

ОДНОМЕРНЫЕ МАССИВЫ

ВАРИАНТ 15

Проверил Выполнил

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

МИНСК 2023

**1** **ЗАДАНИЕ №1**

**1.1 Условие**

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

-количество элементов массива, равных 0;  
- сумму элементов массива, расположенных после минимального элемента.

**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) //проверка

{

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) //При выборе "1" происходит инициализация

{

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

{

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

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

printf("\n");

}

}

else //Ручной ввод элементов

{

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**

В массиве из n элементов удалить каждый k-ый элемент (k=2: 012345 => 024)

**2.2 Алгоритм решения**

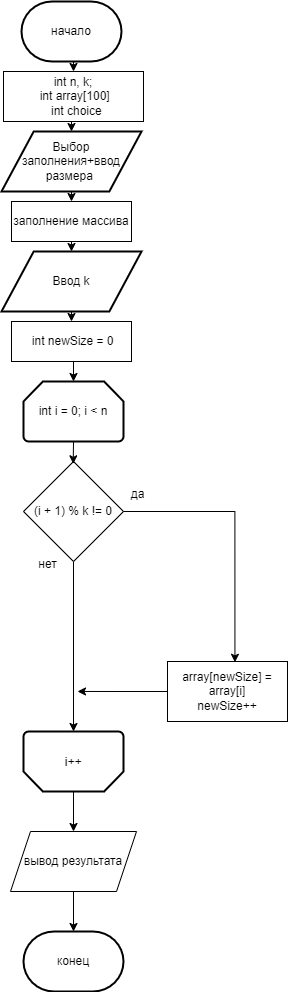


Рисунок 2.1 – Алгоритм задачи 2

**2.3 Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

system("chcp 1251 > nul"); // Установка кодовой страницы консоли для поддержки русских символов

int n, k; // Размер массива и параметр k для удаления элементов

int array[100]; // Массив размером до 100 элементов

printf("Выберите способ ввода элементов массива:\n");

printf("1. Ввести вручную\n");

printf("2. Случайные значения\n");

int choice;

scanf("%d", &choice);

if (choice == 1)

{

printf("Введите размер массива: ");

scanf("%d", &n);

if (n <= 0 || n > 100)

{

printf("Недопустимый размер массива\n");

return 1;

}

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

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

{

printf("Элемент %d: ", i + 1);

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

}

}

else if (choice == 2)

{

printf("Введите размер массива: ");

scanf("%d", &n);

if (n <= 0 || n > 100)

{

printf("Недопустимый размер массива\n");

return 1;

}

srand((unsigned int)time(NULL)); // Инициализация генератора случайных чисел

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

{

array[i] = -50 + rand() % 100; // Заполнение массива случайными значениями

printf("%3d ", array[i]);

}

printf("\n");

}

else

{

printf("Некорректный выбор\n");

return 1;

}

printf("Введите значение k: ");

scanf("%d", &k);

if (k < 1)

{

printf("Значение k должно быть больше или равно 1\n");

return 1;

}

int newSize = 0; // Новый размер массива после удаления элементов

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

{

if ((i + 1) % k != 0) // Сортировка элементов не кратных k

{

array[newSize] = array[i];

newSize++;

}

}

printf("Массив после удаления каждого %d-го элемента:\n", k);

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

{

printf("%3d ", array[i]);

}

return 0;

}

**3 ЗАДАНИЕ №3**

**3.1 Условие**

Для арифметических операций с большими числами, которые не могут быть представлены в памяти компьютера, используется следующий прием. Каждая цифра таких чисел записывается в отдельный элемент массива, и необходимые операции проводятся с элементами массива цифр. Составить программу, выполняющую сложение 30-значных чисел.

**3.2 Алгоритм решения**

Алгоритм предоставлен в «Приложении Б»; Рисунок 1.3

**3.3 Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include <locale.h>

const int NumDigits = 30;

void readNum(char buffer[]) {

printf("Введите число\n");

int success = 1;

while (1) {

scanf("%30s", buffer);

int strLength = strlen(buffer);

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

if (strLength == 30) {

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

if ((buffer[i] < '0') || ('9' < buffer[i])) {

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

success = 0;

break;

}

buffer[i] -= '0';

}

if (success == 1) {

return;

}

}

else {

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

}

}

}

void add(char arr1[], char arr2[], char arrResult[]) {

int carry = 0;

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

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

carry = arrResult[i] / 10;

arrResult[i] %= 10;

}

}

int main() {

setlocale(LC\_ALL, "Rus");

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

readNum(num1);

readNum(num2);

add(num1, num2, sum);

printf("Сумма: ");

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

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

}

printf("\n");

char\* ptr = sum;

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

++ptr;

}

printf("%s\n", ptr);

return 0;

}

**Приложение А**

Блок-схема алгоритма 1

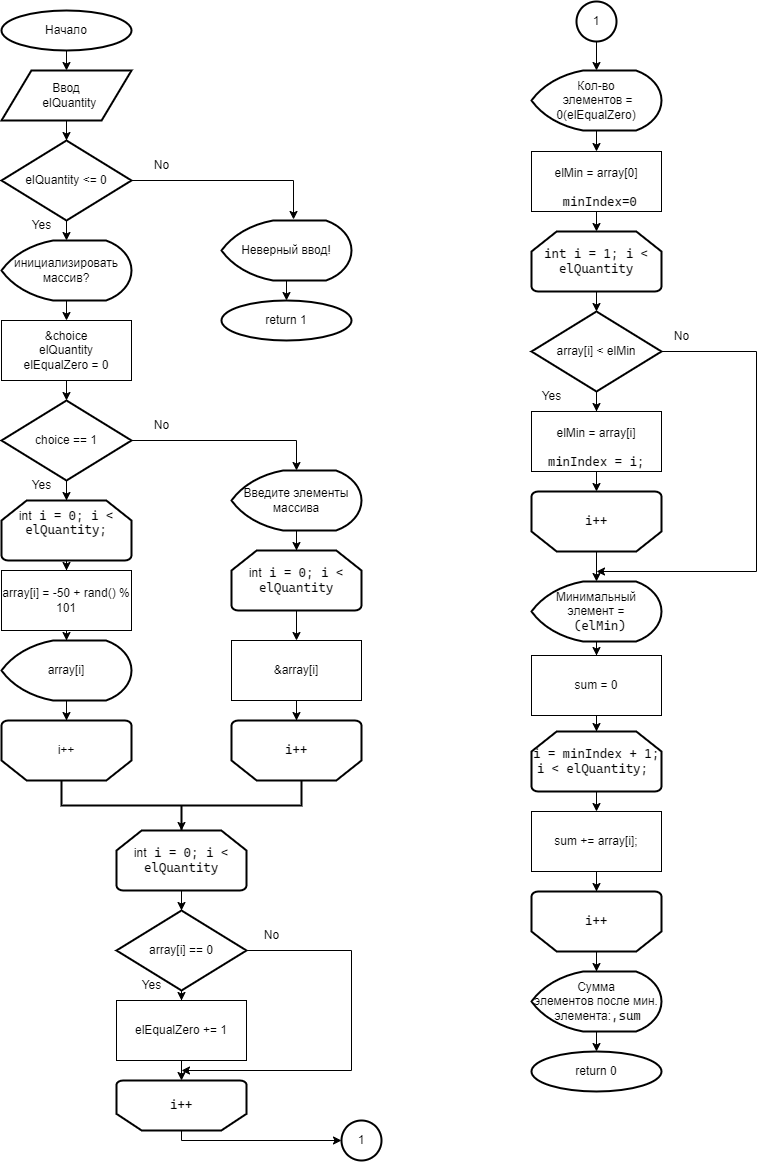


Рисунок 1.1 – Блок-схема №1

**Приложение Б**

Блок-схема алгоритма 3

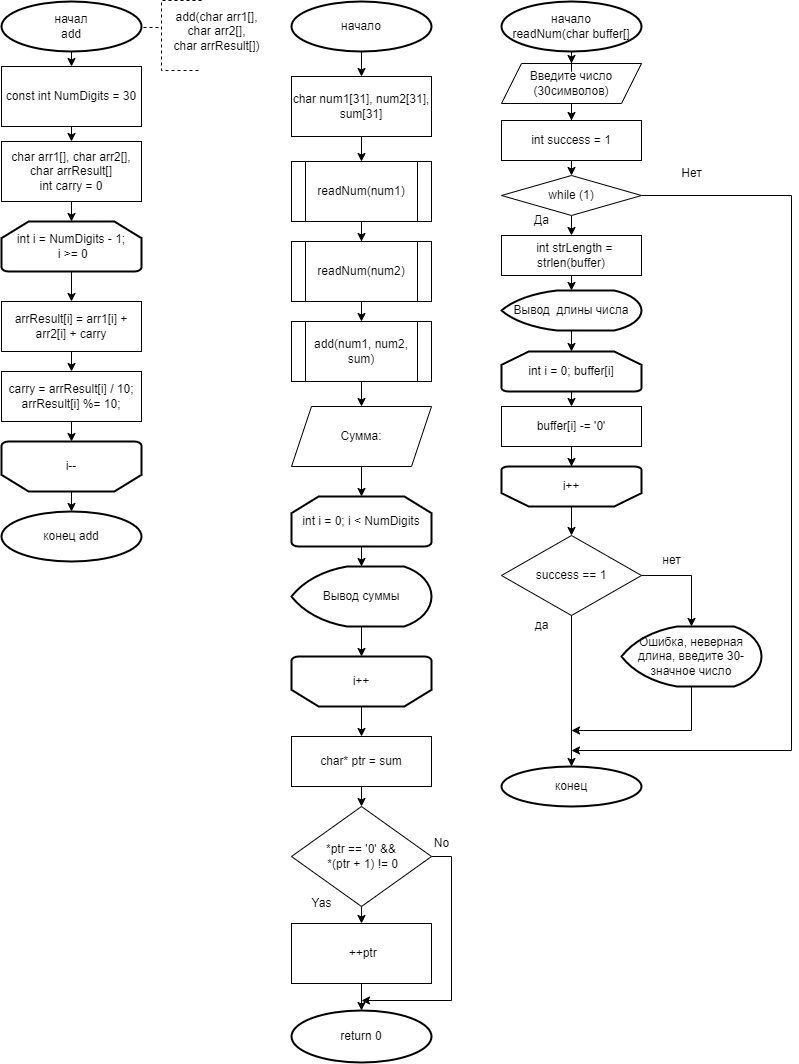


Рисунок 1.3 – Блок-схема №3