ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| к.т.н, доц.. |  |  |  | А.В. Туманова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка числовых последовательностей |
| по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  | 15.03.22 |  | Д.А. Кузнецов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

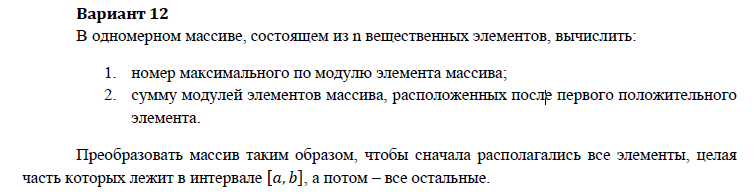
1. **Цель работы**

Целью работы является изучение структуры данных одномерный массив.

1. **Задание**

Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамические, так и нединамические массивы. Размерность последних задаётся именованной константой. Примечание: массивы, созданные с помощью классов (таких, как например, array или vector из библиотеки STL) использовать запрещается.

Примечание: если использовались операторы динамического выделения памяти, то следует вставить дополнительный код, обнаруживающий утечки памяти.

****

1. **Описание созданных функций**

**Имя:** main

**Назначение**: основная функция для ввода/вывода массива и вызова дополнительных функций

**Входные данные:** int: i – счётчик

n – длина массива

double: num – переменная для хранения i-го значения массива

a – левая граница интервала

b – правая граница интервала

double\*: array – динамический одномерный массив

**Побочный эффект:** отсутствует

**Тестовые данные:** n = 6

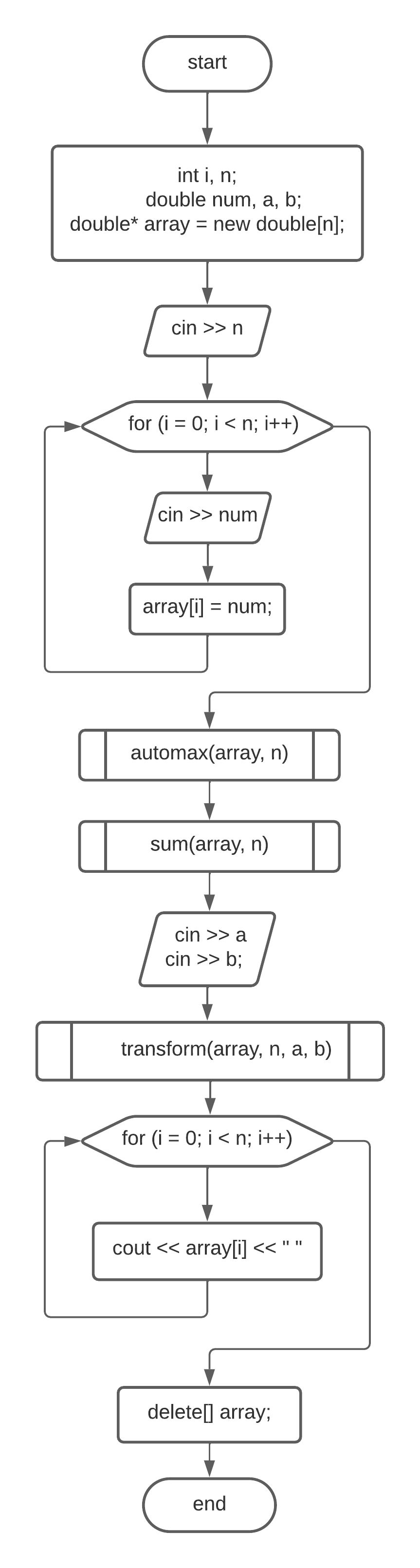
array = [-1; 23; 65; -789; 4; 3]

a = 1

b = 1

**Прототип:** int main()

**Блок-схема:**



1. **Текст программы:**

|  |  |
| --- | --- |
|  |  |
|  |  | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | **int** **main**()  {  setlocale(LC\_ALL, "rus");  **int** i, n;  **double** num, a, b;  std::cout << "Введите длину массива: "; std::cin >> n;  std::cout << "Вводите элементы массива по порядку: **\n**";  **double**\* array = **new** **double**[n];  **for** (i = **0**; i < n; i++)  {  std::cin >> num;  array[i] = num;  }  std::cout << std::endl;  std::cout << "**\n**" << "Номер максимального по модулю элемента: " <<  automax(array, n);  std::cout << "**\n**" << "Cумма модулей, расположенных после первого  положительного элемента: "<< sum(array, n) << std::endl;  std::cout << "Введите интервал (a, b): ";  std::cin >> a; std::cin >> b; std::cout << std::endl;  transform(array, n, a, b);  std::cout << "Преобразованный массив: " << std::endl << " ";  **for** (i = **0**; i < n; i++)  {  std::cout << array[i] << " ";  }  **delete**[] array;  std::cout << std::endl << "**\n**";  // Для обнаружения утечек памяти  \_CrtSetReportMode( \_CRT\_WARN, \_CRTDBG\_MODE\_FILE );  \_CrtSetReportFile( \_CRT\_WARN, \_CRTDBG\_FILE\_STDOUT );  \_CrtSetReportMode( \_CRT\_ERROR, \_CRTDBG\_MODE\_FILE );  \_CrtSetReportFile( \_CRT\_ERROR, \_CRTDBG\_FILE\_STDOUT );  \_CrtSetReportMode( \_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE );  \_CrtSetReportFile( \_CRT\_ASSERT, \_CRTDBG\_FILE\_STDOUT );  \_CrtDumpMemoryLeaks();  } |

**Имя:** automax

**Назначение**: подсчёт номера максимального по модулю элемента массива

**Входные данные:** int: i – счётчик

k – номер максимального по модулю элемента массива

int&: arrsize – размер массива, передаваемый по указателю

double: max – максимальное по модулю значение массива

double\*&: array – динамический одномерный массив

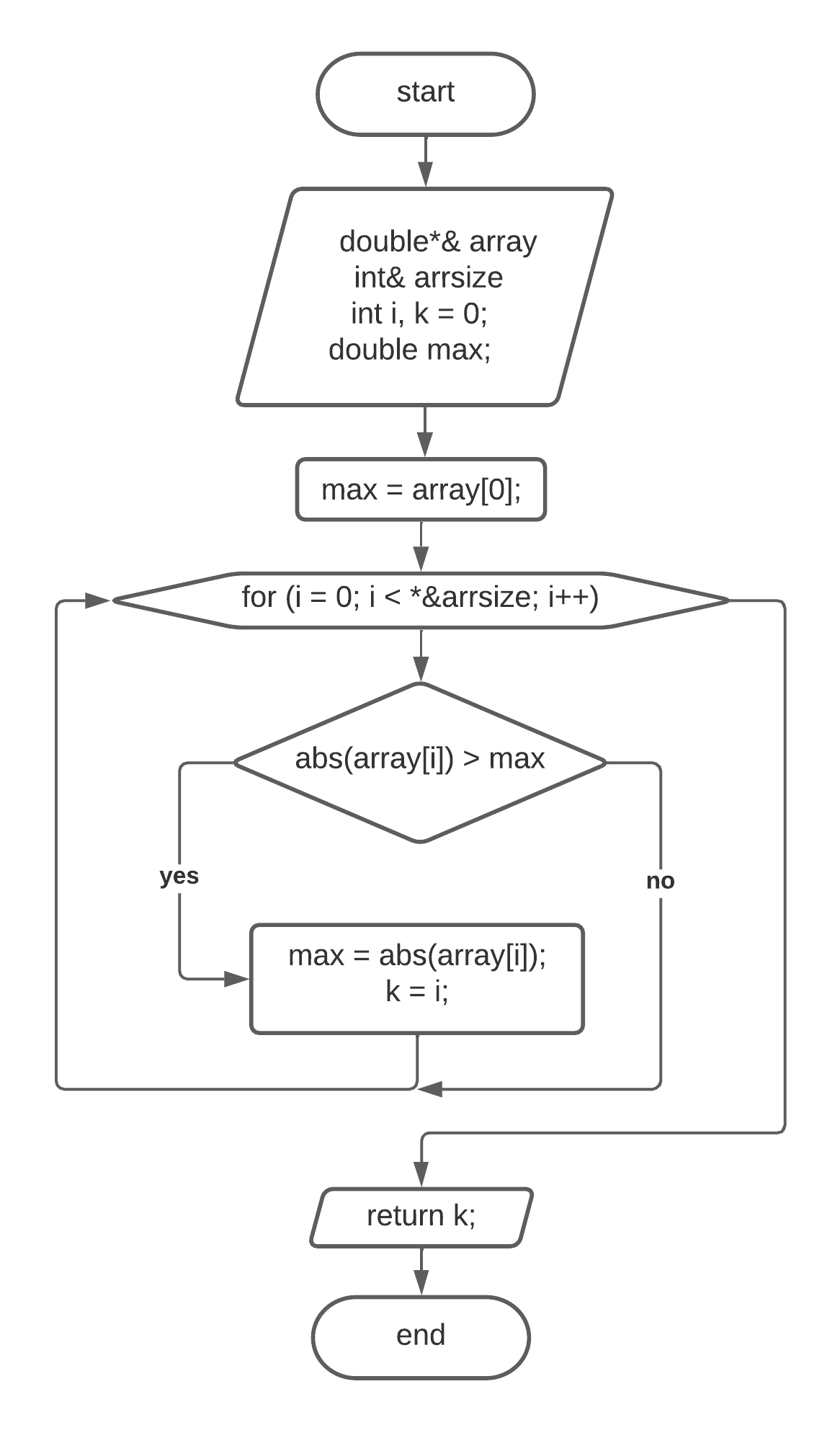
**Побочный эффект:** отсутствует

**Тестовые данные:** array = [-1; 23; 65; -789; 4; 3]

arrsize = 6

**Прототип:** double automax(double\*& array, int& arrsize)

**Блок-схема:**



**Текст программы:**

|  |  |
| --- | --- |
|  |  |
|  |  | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **double** **automax**(**double**\*& array, **int**& arrsize)  {  **int** i, k = **0**;  **double** max;  max = array[**0**];  **for** (i = **0**; i < \*&arrsize; i++)  {  **if** (abs(array[i]) > max)  {  max = abs(array[i]);  k = i;  }  }  **return** k;  } |

**Имя:** sum

**Назначение**: подсчёт суммы элементов массива, располагающихся после первого положительного элемента

**Входные данные:** int: i – счётчик

f – флаг для начала подсчёта

int&: arrsize – размер массива, передаваемый по указателю

double: sum – сумма элементов массива, располагающихся после первого положительного элемента

double\*&: array – динамический одномерный массив

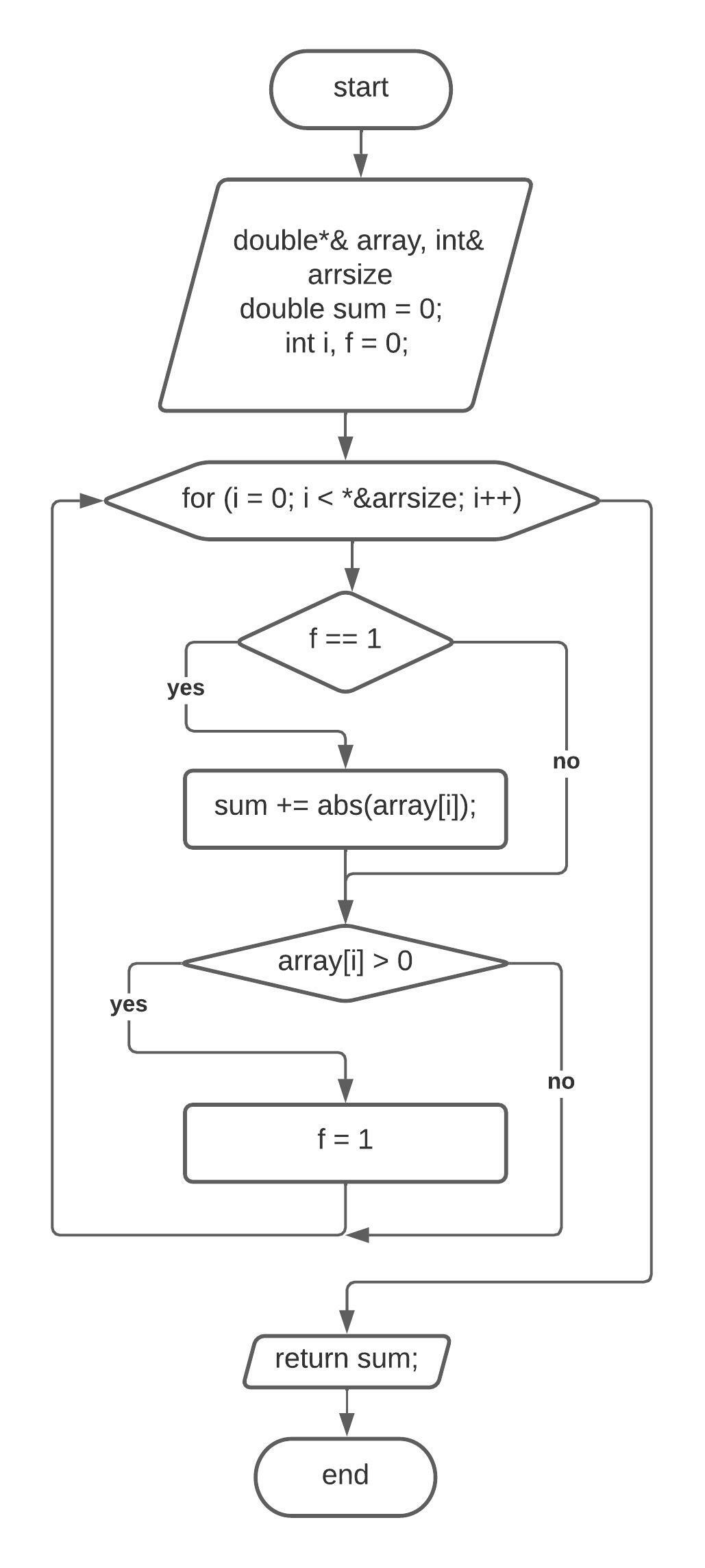
**Побочный эффект:** отсутствует

**Тестовые данные:** array = [-1; 23; 65; -789; 4; 3]

arrsize = 6

**Прототип:** double sum(double\*& array, int& arrsize)

**Блок-схема:**



**Текст программы:**

|  |  |
| --- | --- |
|  |  |
|  |  | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **double** **sum**(**double**\*& array, **int**& arrsize)  {  **double** sum = **0**;  **int** i, f = **0**;  **for** (i = **0**; i < \*&arrsize; i++)  {  **if** (f == **1**)  {  sum += abs(array[i]);  }  **if** (array[i] > **0**)  {  f = **1**;  }  }  **return** sum;  } |

**Имя:** transform

**Назначение**: трансформирование массива по условию задачи

**Входные данные:** int: i – счётчик

k - счётчик

int&: arrsize – размер массива, передаваемый по указателю

double: a – левая граница интервала

b – правая граница интервала

double\*&: array – динамический одномерный массив

double\*: temp – новый, преобразованный массив

**Побочный эффект:** отсутствует

**Тестовые данные:** array = [-1; 23; 65; -789; 4; 3]

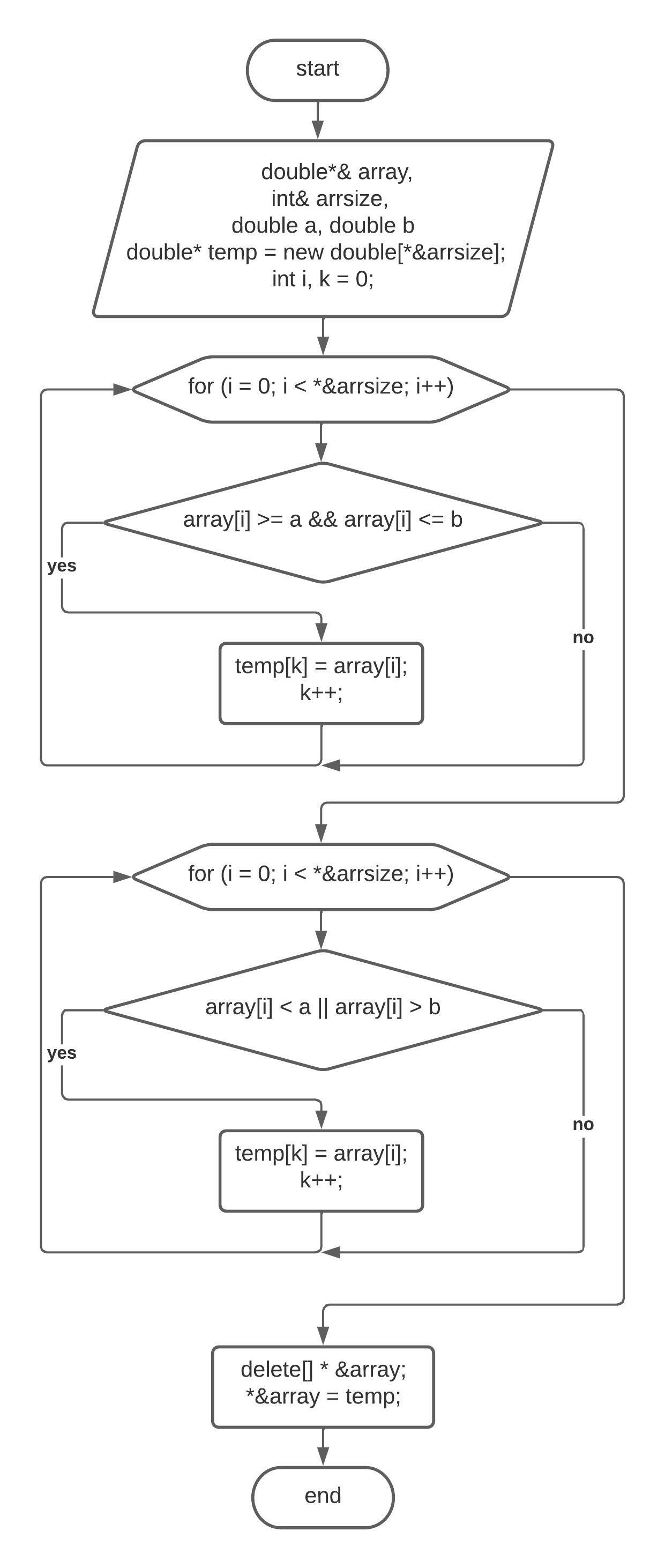
arrsize = 6

a = 1

b = 10

**Прототип:** void transform(double\*& array, int& arrsize, double a, double b)

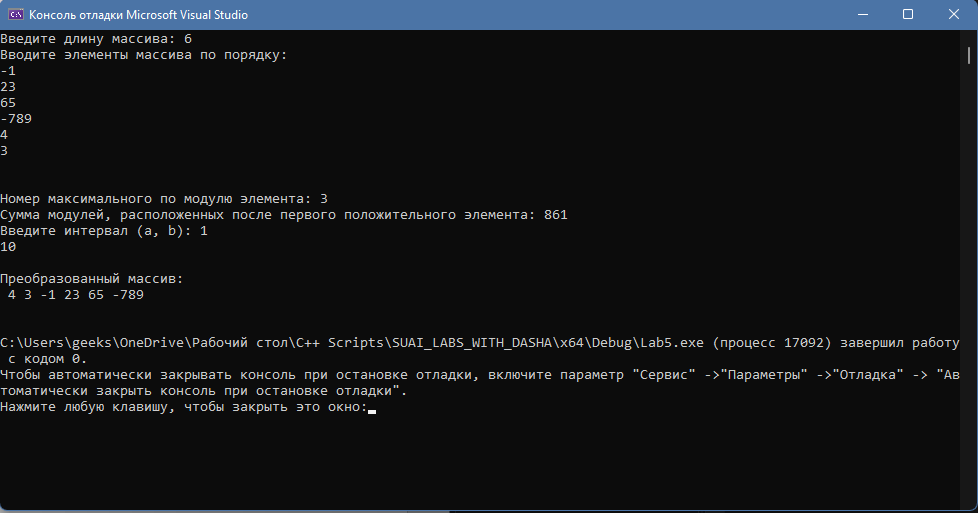
**Блок-схема:**



**Текст программы:**

|  |  |
| --- | --- |
|  |  |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | | | **void** **transform**(**double**\*& array, **int**& arrsize, **double** a, **double** b)  {  **double**\* temp = **new** **double**[\*&arrsize];  **int** i, k = **0**;  **for** (i = **0**; i < \*&arrsize; i++)  {  **if** (array[i] >= a && array[i] <= b)  {  temp[k] = array[i];  k++;  }  }  **for** (i = **0**; i < \*&arrsize; i++)  {  **if** (array[i] < a || array[i] > b)  {  temp[k] = array[i];  k += **1**;  }  }  **delete**[] \* &array;  \*&array = temp; |

**Пример выполнения программы:**

****

1. **Анализ результатов и выводы**

В ходе лабораторной работы я изучил структуру данных одномерный массив