Министерство образования и науки России

Федеральное государственное автономное образовательное

учреждение высшего образования

“Санкт-Петербургский государственный электротехнический

университет им. В. И. Ульянова (Ленина)”

(СПБГЭТУ “ЛЭТИ”)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

**ОТЧЕТ**

**по лабораторной работе № 2 на тему**

**“Одномерные массивы языка С / C++” по**

**дисциплине “Программирование”**

**Выполнил:** *Студент группы 5307 Грицай Н. Д.*

**Проверила:** *к.т.н., доцент Сискович Т.И.*

**Подпись преподавателя:\_\_\_\_\_\_\_\_**

Санкт-Петербург

2015

Оглавление

[Цель работы 2](#_Toc432030262)

[1.Задание 2](#_Toc432030263)

[2.Уточнение задания 2](#_Toc432030264)

[3.Контрольные примеры 2](#_Toc432030265)

[4.Описание переменных 3](#_Toc432030266)

[5.Краткое описание алгоритма 3](#_Toc432030267)

[6.Схема алгоритма 3](#_Toc432030268)

[7.Исходный код программы 4](#_Toc432030269)

[8.Результат 5](#_Toc432030270)

[Вывод 5](#_Toc432030271)

[*Приложение. Схема программы* 6](#_Toc432030272)

# Цель работы

Получить практические навыки использования одномерных массивов языка С / C++.

# 1.Задание

Разработать алгоритм и написать программу, выполняющую многократно по желанию пользователя ввод n элементов одномерного массива, обработку исходного массива, контрольный вывод исходного массива и вывод результата.

# 2.Уточнение задания

Размер массива не превосходит 100 элементов. Пользователь может выбрать один из двух типов обработки массива:

1. в начало массива переставляются все четные элементы, а за ними – нечетные
2. элементы сортируются по возрастанию

# 3.Контрольные примеры

Контрольные примеры представлены в таблице 1.

*Таблица 1. Контрольные примеры*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Исходные данные** | | | **Результат**  **массив newarray** |
| **length** | **массив oldarray** | **sorttype** |
| 1 | 3 | 7 4 8 | 1 | 4 8 7 |
| 2 | 3 | 7 4 8 | 2 | 4 7 8 |
| 3 | 10 | -2 -3 1 4 5 -2 9 8 8 10 | 1 | -2 4 -2 8 8 10 9 -3 5 1 |
| 4 | 10 | -2 -3 1 4 5 -2 9 8 8 10 | 2 | -3 -2 -2 1 4 5 8 8 9 10 |
| 5 | 2 | 1 2 | 1 | 2 1 |
| 6 | 2 | 8 7 | 2 | 7 8 |
| 7 | 4 | -1 -1 4 -2 | 1 | 4 -2 -1 -1 |
| 8 | 4 | -1 -1 4 -2 | 2 | -2 -1 -1 4 |
| 9 | 14 | 10 9 8 7 6 5 4 3 2 1 20 19 18 17 | 2 | 1 2 3 4 5 6 7 8 9 10 17 18 19 20 |

# 4.Описание переменных

Описание переменных представлено в таблице 2.

*Таблица 2. Описание переменных*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| oldarray | int | Исходный массив размера 100 |
| newarray | int | Обработанный массив размера 100 |
| length | short | Размер массива |
| sorttype | short | Тип обработки массива |
| isnewsort, isnewarray | short | Флажки управления циклом |
| i, j | short | Вспомогательные переменные |
| k | int | Вспомогательная переменная |

# 5.Краткое описание алгоритма

1. Ввод размера массива length
2. Ввод элементов массива oldarray
3. Ввод типа обработки массива sorttype
4. Копирование элементов массива oldarray в newarray
5. Если (sorttype == 1), то: обработка массива newarray по четности, иначе: обработка массива newarray по возрастанию
6. Вывод элементов массива oldarray
7. Вывод элементов массива newarray
8. Ввод isnewsort
9. Если(isnewsort != 0), то: переход в пункт 3, иначе: переход в пункт 10
10. Ввод isnewarray
11. Если(isnewarray != 0), то: переход в пункт 1, иначе: выход из программы

# 6.Схема алгоритма

Схема алгоритма представлена в приложении 1

# 7.Исходный код программы

#include "stdafx.h"

#include "iostream"

#include "Windows.h"

#define ARRAYMAXSIZE 100

int main()

{

short isnewarray, isnewsort, sorttype, length, i, j;

int oldarray[ARRAYMAXSIZE], newarray[ARRAYMAXSIZE], k;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

do

{

system("cls");

//Ввод массива

printf\_s("Введите размер массива: ");

do

{

scanf\_s("%hi", &length);

if ((length > ARRAYMAXSIZE) || (length < 1))

printf\_s("Ошибка! Размер массива не может быть меньше единицы

и больше %d.\nВведите размер массива

еще раз: ", ARRAYMAXSIZE);

} while ((length > ARRAYMAXSIZE) || (length < 1));

printf\_s("\nВвод элементов массива\n");

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

{

printf\_s("Введите %hi-й элемент массива: ", i+1);

scanf\_s("%i", oldarray+i);

}

printf\_s("\nВвод завершен\n");

//

do

{

printf("Выберите тип обработки исходного массива(По четности - 1/По

возрастанию - 2): ");

do

{

scanf\_s("%hi", &sorttype);

if ((sorttype != 1 ) && (sorttype != 2))

printf\_s("Ошибка! Введите существующий тип обработки:

");

} while ((sorttype != 1) && (sorttype != 2));

//копирование массивов

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

newarray[i] = oldarray[i];

//

if (sorttype == 1)

{

//обработка массива по четности

j = 0;

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

if (!(newarray[i] & 1))

{

k = newarray[i];

newarray[i] = newarray[j];

newarray[j] = k;

j++;

}

}

else

{

//обработка массива по возрастанию

for (i = length - 1; i > 0; i--)

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

if (newarray[j] > newarray[j + 1])

{

k = newarray[j];

newarray[j] = newarray[j + 1];

newarray[j + 1] = k;

}

}

//Вывод исходного массива

printf\_s("\nИсходный массив:\n");

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

printf\_s("%5i ", oldarray[i]);

printf\_s("\n");

//Вывод обработанного массива

printf\_s("\nОбработанный массив:\n");

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

printf\_s("%5i ", newarray[i]);

printf\_s("\n");

//

printf\_s("\nПрименить другой тип сортировки?(Да-1/Нет-0): ");

scanf\_s("%hi", &isnewsort);

} while (isnewsort);

printf\_s("Ввести новый массив?(Да-1/Нет-0): ");

scanf\_s("%hi", &isnewarray);

} while (isnewarray);

return 0;

}

# 8.Результат

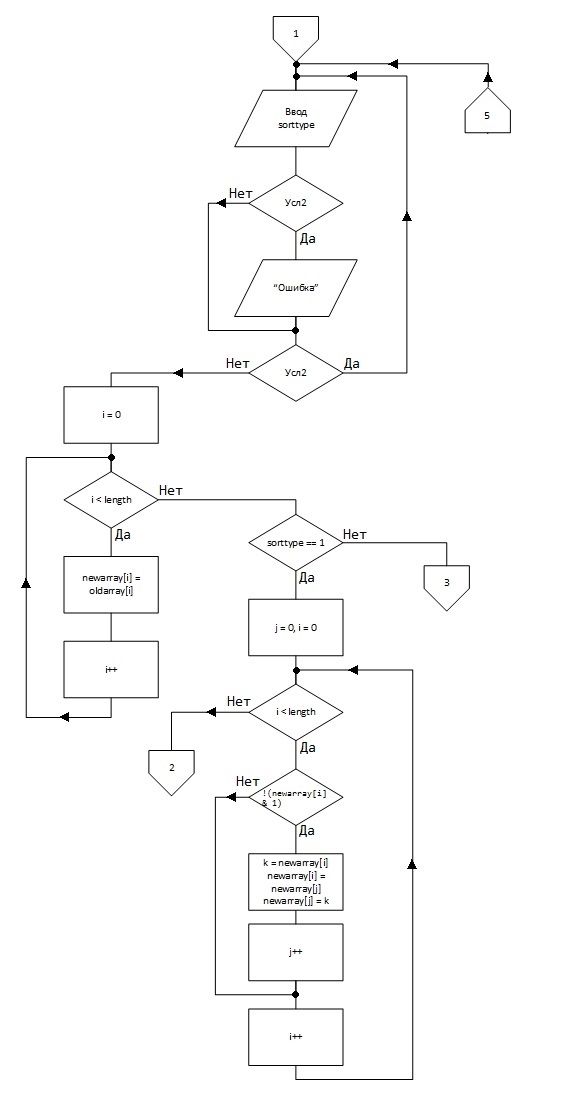
При выполнении программы получены результаты, совпадающие с контрольными примерами, представленными в таблице 1. Ошибки не обнаружены.

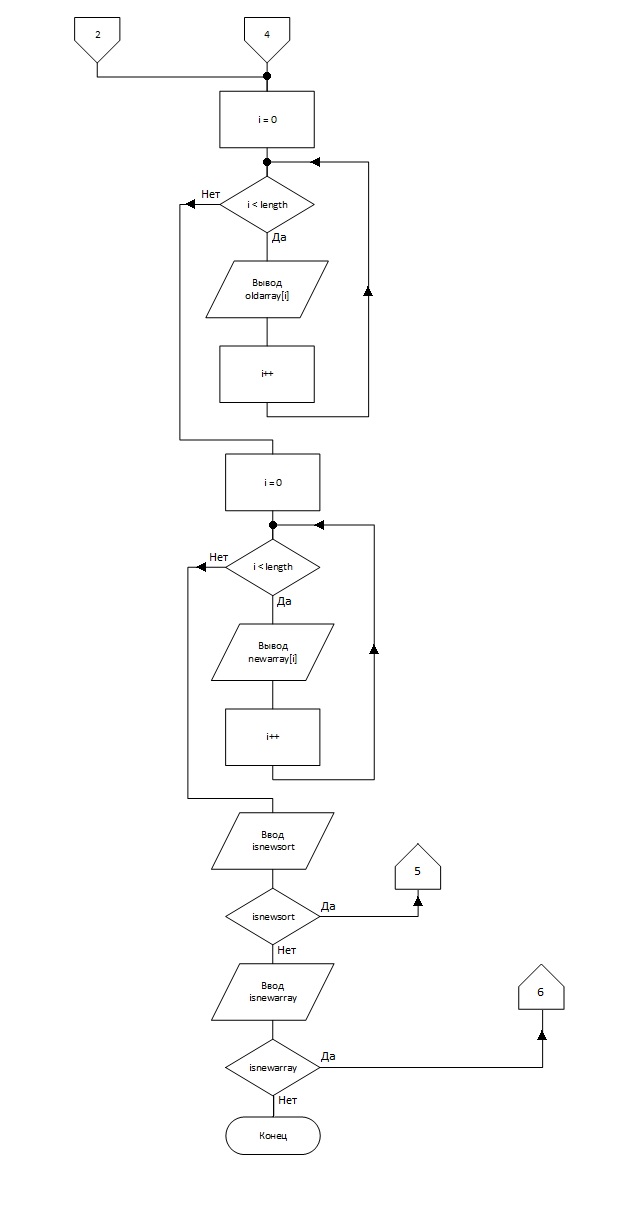
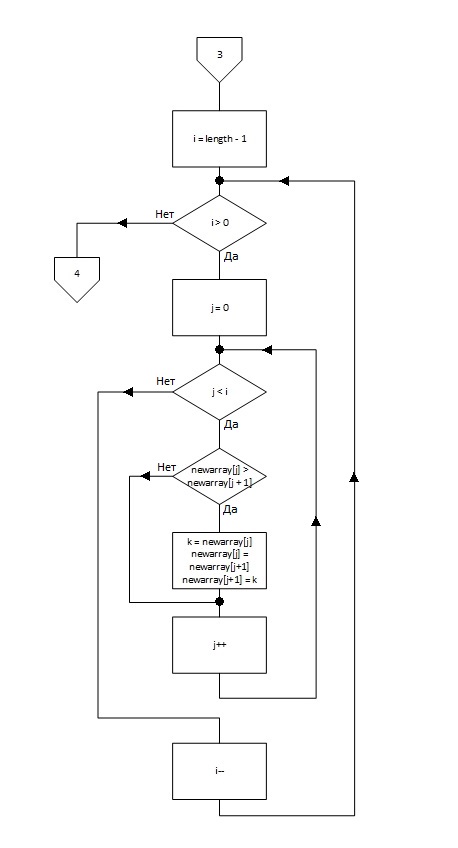
# Вывод

В ходе выполнения работы были получены практические навыки использования одномерных массивов языка С / C++.

# *Приложение. Схема программы*

# 

**

**

*Рисунок 1. Схема программы*

Усл1: (length > 100) || (length < 1)

Усл2: (sorttype != 1) && (sorttype != 2)