*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)*** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА Компьютерные Системы и сети (ИУ6)

**Отчет**

**по лабораторной работе №4**

**варианта №7**

**Название лабораторной работы:**

Одномерные массивы

**Дисциплина:**

Основы программирования

Студент гр. **ИУ6-12** 10/11/16**\_\_\_\_\_\_\_\_\_ Векшин Роман**

(Подпись, дата) (И.О. Фамилия)

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черноусова Татьяна Геннадьевна**

(Подпись, дата) (И.О. Фамилия)

**Задание:**

Дан целочисленный массив Н(30). Оставить его без изменения, если он упорядочен по невозрастанию или по неубыванию. В противном случае удалить из него все элементы, нарушающие регулярность (по усмотрению автора регулярностью считать невозростание или неубывание).

**Схема алгоритма:**

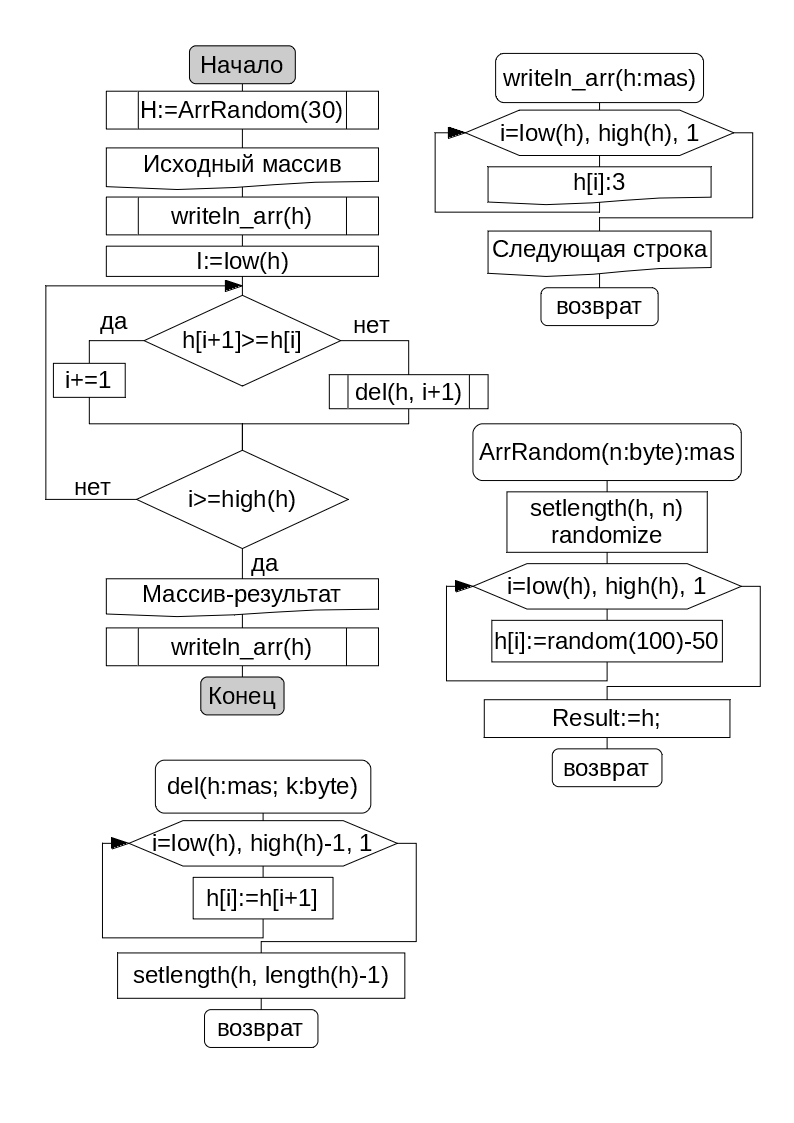


Рисунок -Схема алгоритма

**Код программы:**

program project1;

{$APPTYPE CONSOLE}

type

mas = array of integer;

**//Процедура удаления элемента с индексом k из массива со сдвигом "хвоста"**

procedure del(var h: mas; const k: byte);

var

i: byte;

begin

for i := k to high(h) - 1 do

h[i] := h[i + 1];

setlength(h, length(h) - 1);

end;

**//Процедура вывода массива**

procedure writeln\_arr(const h: mas);

var

i: byte;

begin

for i := low(h) to high(h) do

Write(h[i]: 3);

writeln();

end;

**//Процедура заполнения массива**

function ArrRandom(const n: byte): mas;

var

i: byte;

h: mas;

begin

setlength(h, n);

randomize;

for i := low(h) to high(h) do

h[i] := random(100) - 50;

Result := h;

end;

var

h: mas;

i: byte;

begin

h := ArrRandom(30);

writeln(utf8toansi('Исходный массив:'));

writeln\_arr(h);

i := low(h);

repeat

if h[i + 1] >= h[i] then

i := i + 1

else

del(h, i + 1);

until i >= high(h);

writeln(utf8toansi('Массив-результат:'));

writeln\_arr(h);

readln();

end.

**Тестирование:**

Таблица -Результаты тестирования программы

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Полученный результат |
| -47 -34 -2 22 -31 23 5 -38 21 -27 -48 13 -17 -21 -12 16 -30 22 -42 21 6 -26 1 29 3 -4 10 -37 9 -45 | -47 -34 -2 22 23 29 | -47 -34 -2 22 23 29 |
| 37 8 -46 47 -44 -28 -13 -22 37 -48 0 12 35 31 0 -13 -45 3 -41 4 43 -49 -29 -27 -39 -19 -48 35 -19 -16 | 37 47 | 37 47 |
| 20 -15 46 -34 -31 40 -34 -46 -11 25 41 45 5 -2 -45 2 -28 -17 -15 23 43 47 -40 0 -40 -43 39 5 28 -11 | 20 46 47 | 20 46 47 |
| 12 -35 6 19 -10 -17 -37 -18 5 17 37 -24 23 -41 9 0 20 -4 -15 -2 36 8 19 -7 41 44 -49 -42 -1 -3 | 12 19 37 41 44 | 12 19 37 41 44 |

**Пример работы программы:**

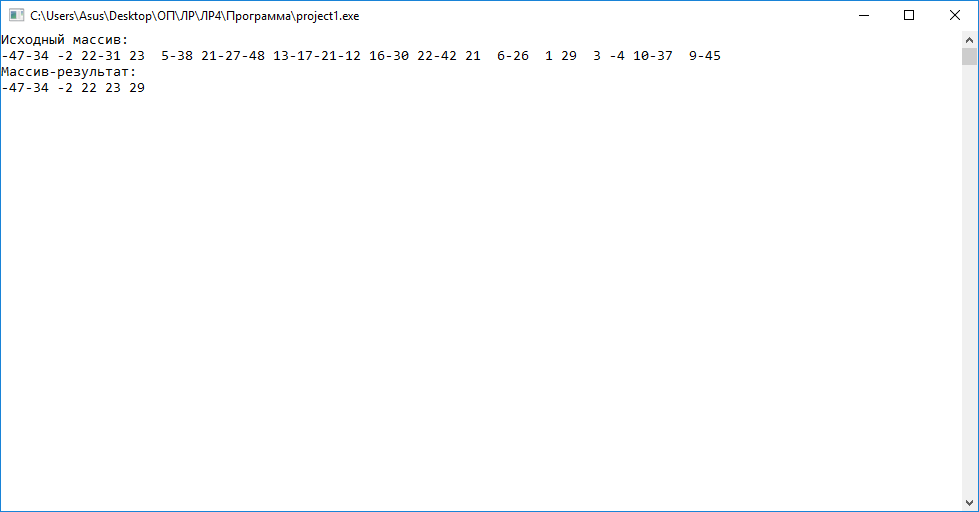


Рисунок -Пример работы программы

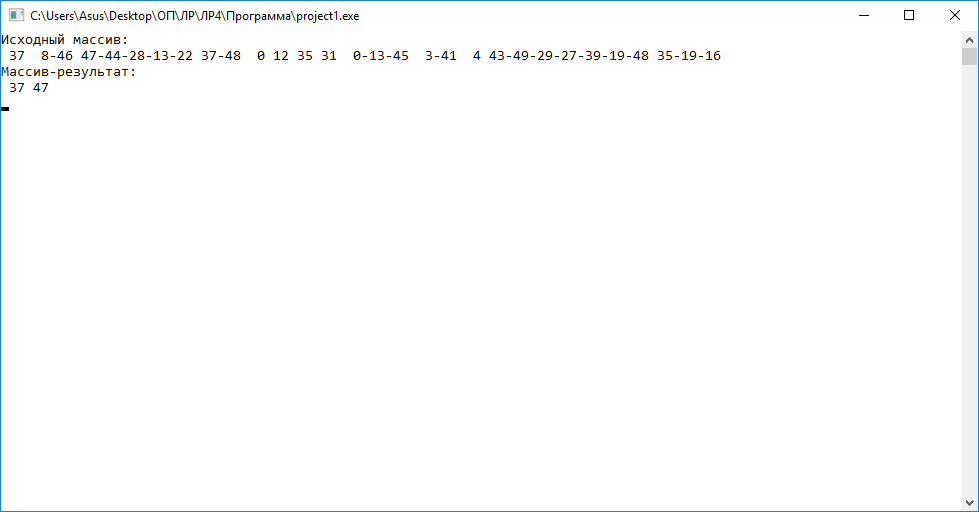


Рисунок -Пример работы программы

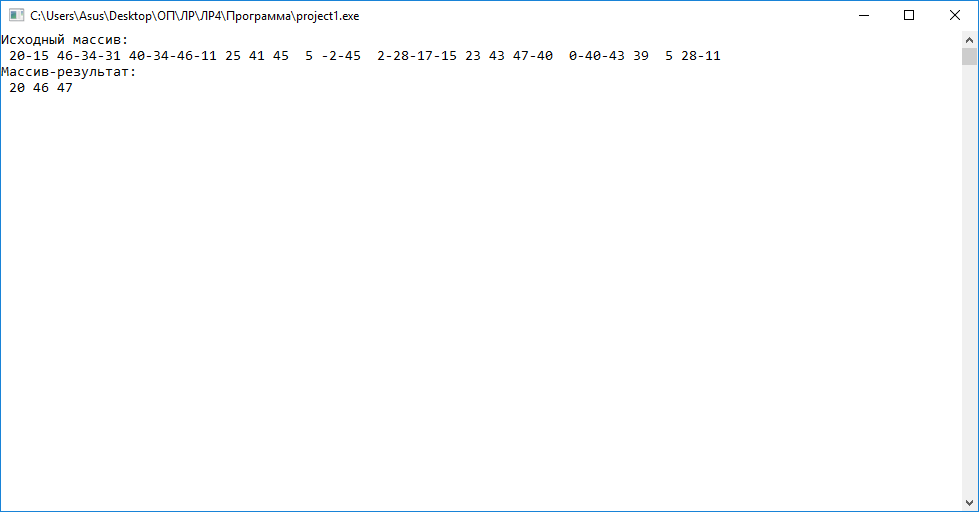


Рисунок -Пример работы программы

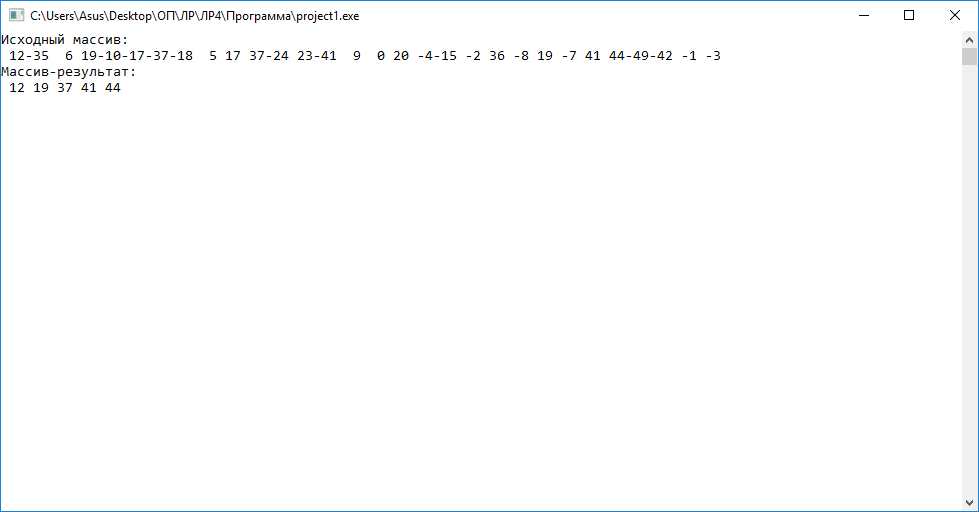


Рисунок -Пример работы программы

**Вывод:**

1. Разработан алгоритм для упорядочивания массива в порядке нестрогого возрастания(неубывания), и составлена его схема в среде LibreOffice Draw(см. рис. 1).
2. Создан код программы по схеме алгоритма в среде Lazarus
3. Проведено тестирование (см. табл. 1).
4. Тестирование показало корректность работы программы (см. рис. 2-5)