**МИНОБРНАУКИ РОССИИ**

**ФГБОУ ВПО «Пензенский государственный университет»**

**Кафедра «Математическое обеспечение и применение ЭВМ»**

**ОТЧЕТ**

по лабораторной работе №12

«Указатели, стеки, очереди»

Выполнил: гр. 19ВИ1

Мельхов. А. А.

Проверил: к.т.н., доцент

Казаков Б.В.

Пенза 2019 г.

**Лабораторная работа №12**

**Тема: «**Указатели, стеки, очереди**»**

Вариант №16

**Цели**: научиться работать с динамической и статической памятью

**Общее задание**

Требуется разработать две программы, выполняющие следующие действия:

1. Ввод размера массива (или двух – в зависимости от задания)

2. Выделения памяти под исходный массив (массивы)

3. Ввод исходного массива (массивов)

4. Вывод введенных массивов

5. Обработка массива (массивов) в соответствии с вариантом

6. Вывод получившихся массивов

7. Освобождение динамической памяти

**Замечания:**

1) Обе программы делают одно и то же, но в первой программе все переменные и массивы должны быть размещены статически т.е. без использования динамической памяти, а во второй программе все используемые массивы, как основные, так и вспомогательные, должны выделяться динамически.

2) Количество элементов в исходных массивов до 20 штук.

3) Элементами массивов являются целые числа.

4) Множества в программах не использовать.

5) После каждого изменения массивов новое состояние необходимо вывести на экран.

6) **«Скопировать элементы»** – элементы из исходного массива добавляются в результирующий массив.

7) **«Перенести элементы»** – элементы из исходного массива добавляются в результирующий массив, после чего удаляются из исходного.

**Задание:** Ввести массив А. В массив В скопировать все элементы массива А, имеющие четный индекс и четное значение. Массив В отсортировать по убыванию, используя модифицированный метод простого выбора.

**Коды программ на языке Pascal:**

Статические переменные

**var**

ar, dublAr: **array** [1..20] **of** integer;

i, j, dI, N, st\_min, st, id\_min: integer;

**begin**

write('Введите кол-во элементов в массиве: ');

readln(N);

**for** i:=1 **to** N **do**

**begin**

readln(ar[i]);

**end**;

dI:= 0;

write('Исходный массив: ');

**for** i:=1 **to** N **do**

**begin**

write(ar[i], ' ');

**if**  **then**

**begin**

dI+=1;

dublAr[dI]:= ar[i]

**end**;

**end**;

writeln;

write('Массив B: ');

**for** i:=1 **to** dI **do**

write(dublAr[i], ' ');

writeln;

**for** i := 1 **to** dI - 1 **do**

**begin**

st\_min := dublAr[i];

id\_min := i;

**for** j := i + 1 **to** dI **do**

**if** dublAr[j] < st\_min **then begin**

st\_min := dublAr[j];

id\_min := j;

**end**;

st := dublAr[i];

dublAr[i] := dublAr[id\_min];

dublAr[id\_min] := st;

**end**;

write('Массив B после сортировки: ');

**for** i:=1 **to** dI **do**

write(dublAr[i], ' ');

**end**.

**Динамические переменные**:

**var**

ar, dublAr: **array of** integer;

dI, N, st\_min, st, id\_min: integer;

**begin**

write('Введите кол-во элементов в массиве: ');

readln(N);

setLength(ar, N);

**for var** i:=0 **to** N-1 **do**

**begin**

readln(ar[i]);

**end**;

setLength(dublAr, N);

dI:= 0;

write('Исходный массив: ');

**for var** i:=0 **to** High(ar)-1 **do**

**begin**

write(ar[i], ' ');

**if** (i **mod** 2 = 0) **and** (ar[i] **mod** 2 = 0) **then**

**begin**

dI+=1;

dublAr[dI-1]:= ar[i]

**end**;

**end**;

writeln;

write('Массив B: ');

**for var** i:=0 **to** dI-1 **do**

write(dublAr[i], ' ');

writeln;

**for var** i := 0 **to** dI - 2 **do**

**begin**

st\_min := dublAr[i];

id\_min := i;

**for var** j := i + 1 **to** dI-1 **do**

**if** dublAr[j] < st\_min **then begin**

st\_min := dublAr[j];

id\_min := j;

**end**;

st := dublAr[i];

dublAr[i] := dublAr[id\_min];

dublAr[id\_min] := st;

**end**;

write('Массив B после сортировки: ');

**for var** i:=0 **to** dI-1 **do**

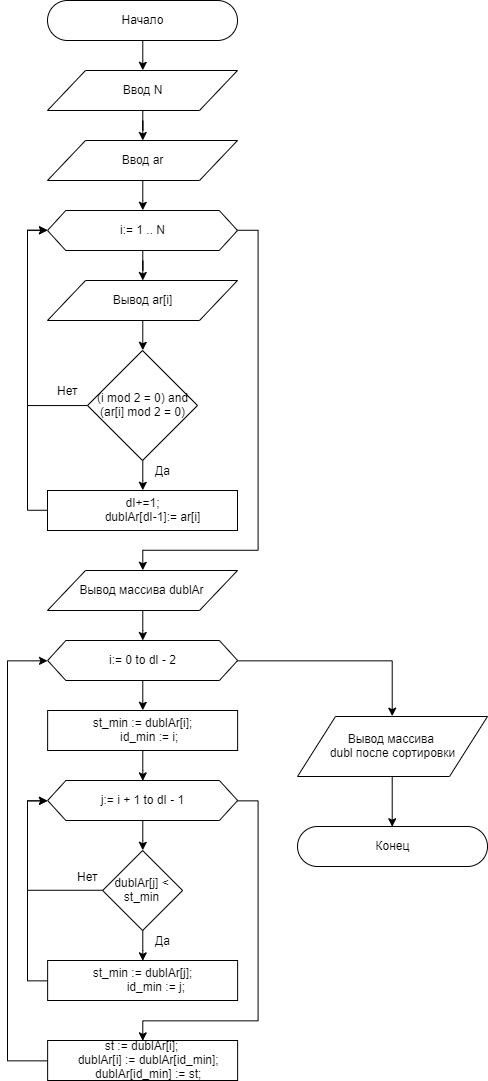
write(dublAr[i], ' ');

dublAr:= nil;

ar:= nil;

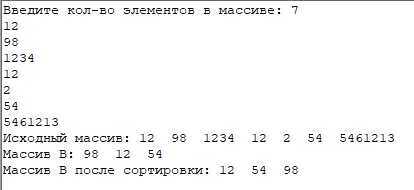
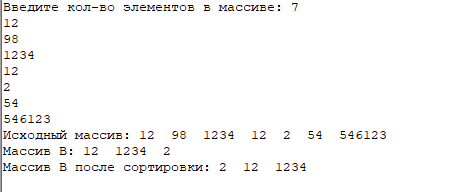
**end**.

**Блок-схема алгоритма работы программы:**



**Результаты работы программы:**

1 программа 2 программа

**Вывод:**

* Научился работать с динамической и статической памятью