

Лабораторная работа №10

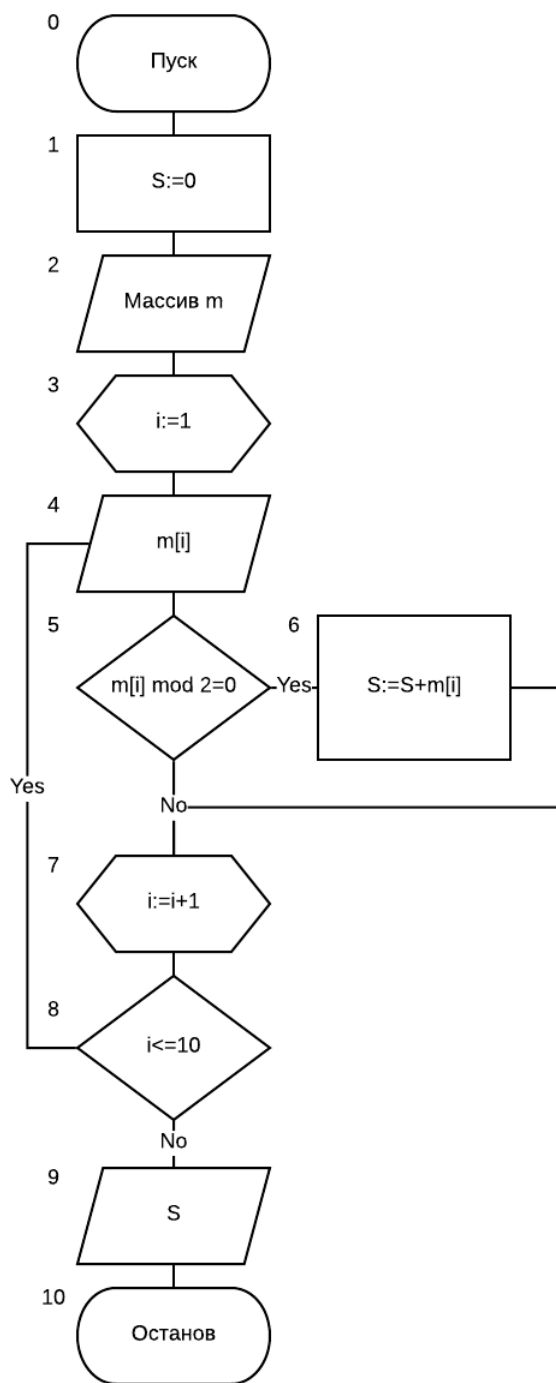
Итерационные ЦВП с управлением по индексу и функции.

Вариационный ряд.

2. Цель: научиться создавать и применять Итерационные циклические вычислительные процессы с управлением по индексу и функции.
3. Оборудование: ПК, Lazarus, PascalABC.

Задание №1

4. Дан одномерный массив. Найти сумму четных (по значению) элементов массива.
5. $m[i] \bmod 2$
- 6.



7.

Имя	Смысл	Тип
i	Индекс массива	integer
S	Сумма	integer
m	массив	Array of integer

8.

```

1  program Project1;
.   var m: array [1..10] of integer;
.   i, S: integer;
.   begin
5   S:=0;
.   randomize;
.   for i:=1 to 10 do
.   begin
.       m[i]:=random(100);
10  write (m[i], ' ');
.       if m[i] mod 2=0 then
.           S:=S+m[i];
.       end;
.       writeln ('Summa chetnih elementov = ', S);
15  readln;
.   end.

```

9. 44 14 4 73 8 54 81 32 7 69 Summa chetnih elementov = 156

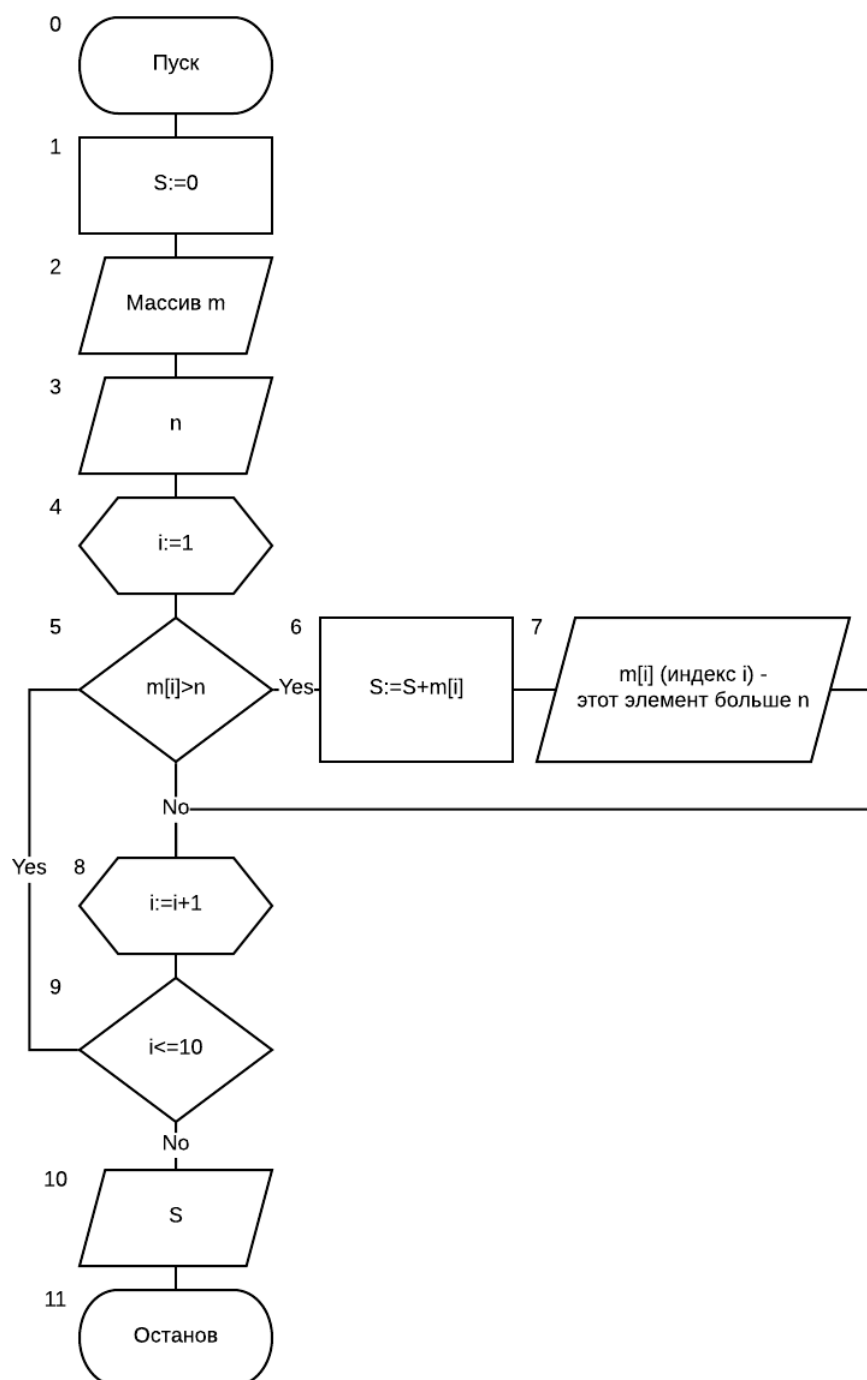
10. Чтобы определить, какие элементы массива являются чётными, программа определяет остаток после целочисленного деления на 2. Если остаток равен нулю, то этот элемент добавляется в сумму.

Задание №2

4. Дан одномерный массив. С клавиатуры вводится число. Найти сумму всех элементов массива, значение которых больше числа, введенного с клавиатуры и вывести их индексы.

5. $m[i] > n$ $S := S + m[i]$

6.



7.

Имя	Смысл	Тип
i	Индекс массива	integer
S	Сумма	integer
n	Вводимое число	integer
m	массив	Array of integer

8.

```

1  program Project1;
.   var m: array [1..10] of integer;
.   i, S, n: integer;
.   begin
5   writeln ('Vvedite chislo');
.   readln (n);
.   S:=0;
.   randomize;
.   for i:=1 to 10 do
10  begin
.   m[i]:=random(100);
.   if m[i]>n then
.   begin
.   S:=S+m[i];
15  writeln (m[i], ' (indecs ', i, ') - etot element bolshe ', n);
.   end;
.   end;
.   writeln ('Summa = ', S);
.   readln;
20 end.

```

```

Vvedite chislo
50
88 (indecs 2) - etot element bolshe 50
60 (indecs 5) - etot element bolshe 50
92 (indecs 7) - etot element bolshe 50
67 (indecs 9) - etot element bolshe 50
Summa = 307

```

9.

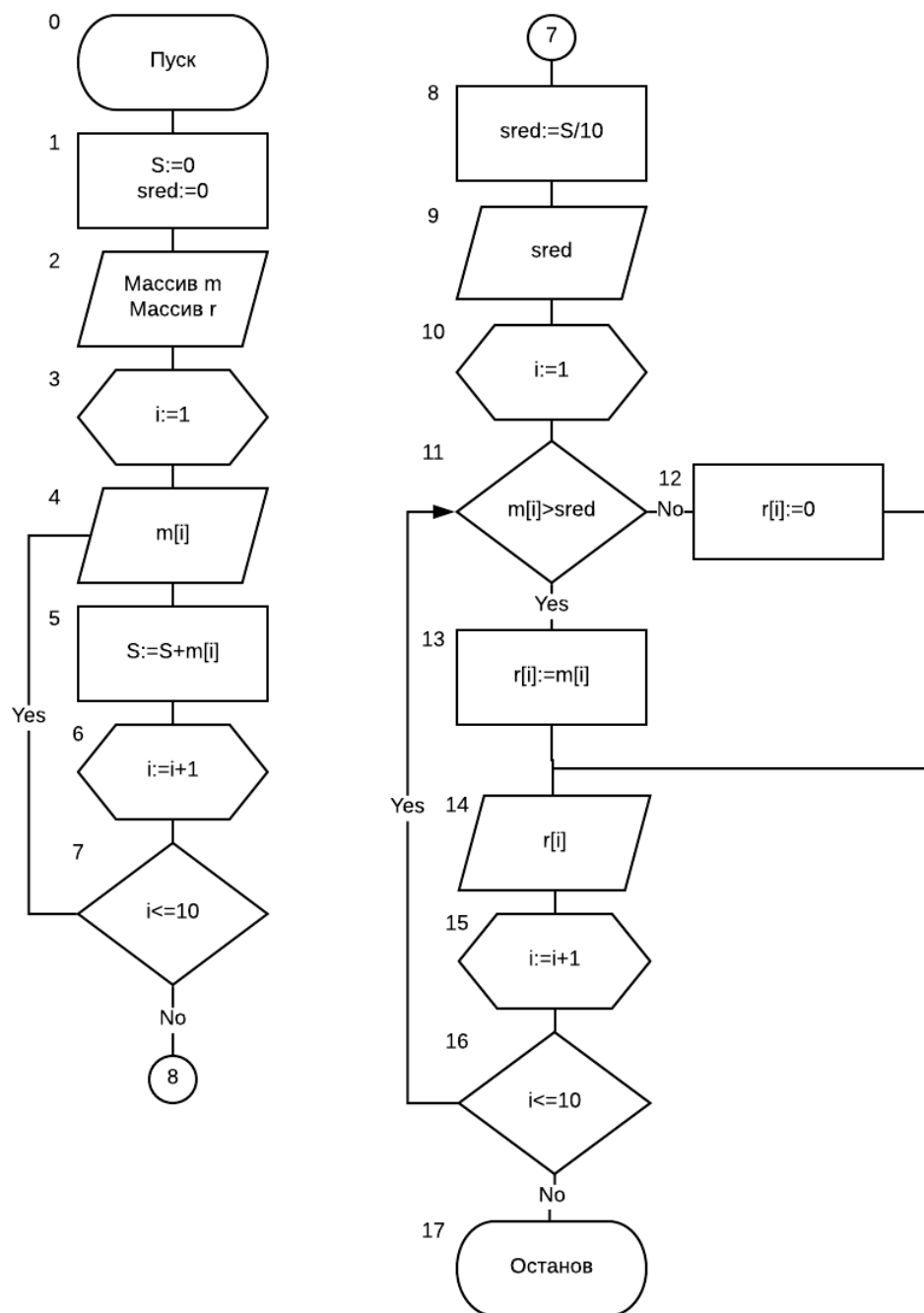
10. На входе в цикл проверяется, является ли этот элемент массива больше числа, введённого пользователем. Если является, то этот элемент прибавляется к накопителю суммы и выводится на экран вместе со своим индексом.

Задание №3

4. Дан одномерный массив. Найти его среднее арифметическое. Составить второй массив, элементами которого будут элементы первого массива, которые больше среднего арифметического. Остальные элементы заменить нулями.

5. $S := S + m[i]$ $sred := S / 10$

6.



7.

Имя	Смысл	Тип
m	начальный массив	Array of integer
r	Получившийся массив	Array of integer
S	Накопитель суммы	Integer
i	Индекс массивов	Integer
sred	Среднее арифметическое массива m	Real

8.

```

program a3;
var
  m: array [1..10] of integer;
  r: array [1..10] of integer;
  S,i: integer;
  sred: real;
begin
  randomize;
  S:=0;
  sred:=0;
  for i:=1 to 10 do
    begin
      m[i]:=random(100);
      write (m[i], ' ');
      S:=S+m[i];
    end;
  sred:=S/10;
  writeln ('Среднее арифметическое массива m = ',sred);
  for i:=1 to 10 do
    begin
      if m[i]>sred then r[i]:=m[i]
      else r[i]:=0;
      write (r[i], ' ');
    end;
  end.

```

9.

Окно вывода

2	34	63	47	85	5	18	13	55	45	Среднее арифметическое массива m = 36.7
0	0	63	47	85	0	0	0	55	45	

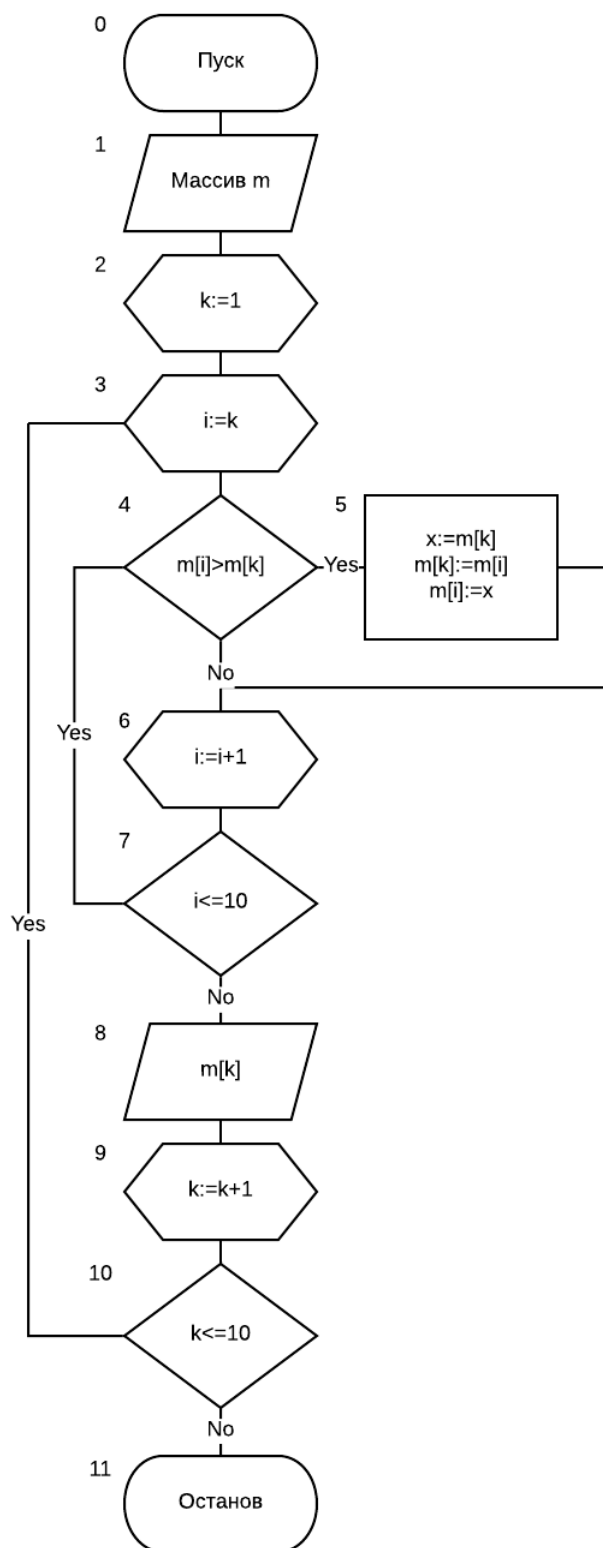
10. Первый цикл направлен на суммирование всех элементов массива m. После массива в переменную sred записывается среднее арифметическое массива m. После этого во втором цикле элементу массива r присваивается то же значение, что и у элемента массива m с тем же индексом, если этот элемент массива m больше среднего арифметического массива m. Если же этот элемент меньше, то элементу массива r присваивается значение 0.

Задание №4

4. Дан одномерный массив. Упорядочить его по убыванию.

5. $x:=m[k]$ $m[k]:=m[i]$ $m[i]:=x$

6.



7.

Имя	Смысл	Тип
m	массив	Array of integer
i	Индекс/параметр внутреннего цикла	integer

k	Индекс/параметр внешнего цикла	integer
x	Промежуточная переменная (запоминает значение m[k])	integer

8.

```

program m4;
var
  m: array [1..10] of integer;
  i,k,x: integer;
begin
  randomize;
  for i:=1 to 10 do
  begin
    m[i]:=random(100);
    write (m[i], ' ');
  end;
  writeln ('- изначальный вид массива');
  for k:=1 to 10 do
  begin
    for i:=k to 10 do
      if m[i]>m[k] then
      begin
        x:=m[k];
        m[k]:=m[i];
        m[i]:=x;
      end;
    write (m[k], ' ');
  end;
  writeln ('- итоговый вид массива');
end.

```

Окно вывода

```

47  67  61  20  29  35  17  58  1  22  - изначальный вид массива
67  61  58  47  35  29  22  20  17  1  - итоговый вид массива

```

9.

10. Для упорядочивания элементов массива по убыванию программа использует массив в массиве. Внутренний массив перебирает все элементы от к-того до последнего. Если какой-то элемент больше к-того, то алгоритм меняет их местами. Таким образом наибольшее число окажется правее. Внешний цикл перебирает значения к от 1 до 10 и после выполнения внутреннего цикла выводит к-ый элемент.

11. Вывод: Итерационные циклические вычислительные процессы с управлением по индексу и функции отлично подходят для различных работ с элементами массива: будь то нахождение суммы определённых элементов или же сортировка по определённому признаку.