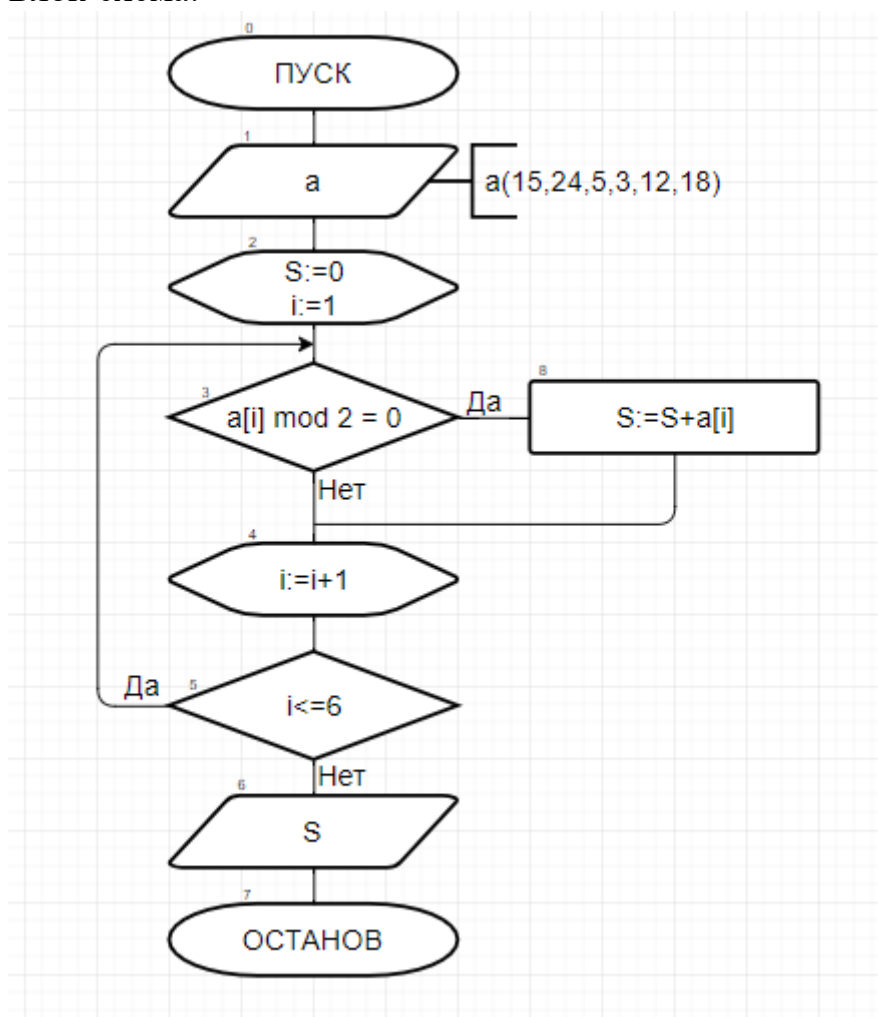


## Лабораторная работа № 10

1. Тема лабораторной работы: итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: изучение ИЦВП с управлением по индексу и функции, вариационного ряда с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: найти сумму четных (по значению) элементов одномерного массива.
5. Математическая модель:  
Если  $a[i] \bmod 2 = 0$ , тогда  $S = S + a[i]$ , иначе переход к следующему элементу массива.
6. Блок-схема:



7. Список идентификаторов:

Имя	Тип	Смысл
a	integer	Массив с числами
S	integer	Сумма четных по значению элементов массива
i	integer	Индекс массива

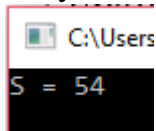
8. Код программы:  
program zadanie1;

```

const
  a: array [1..6] of integer = (15,24,5,3,12,18);
var
  S, i: integer;
begin
  S:=0;
  for i:=1 to 6 do
    if a[i] mod 2 = 0 then
      S:=S+a[i];
  writeln('S = ', S);
  readln();
end.

```

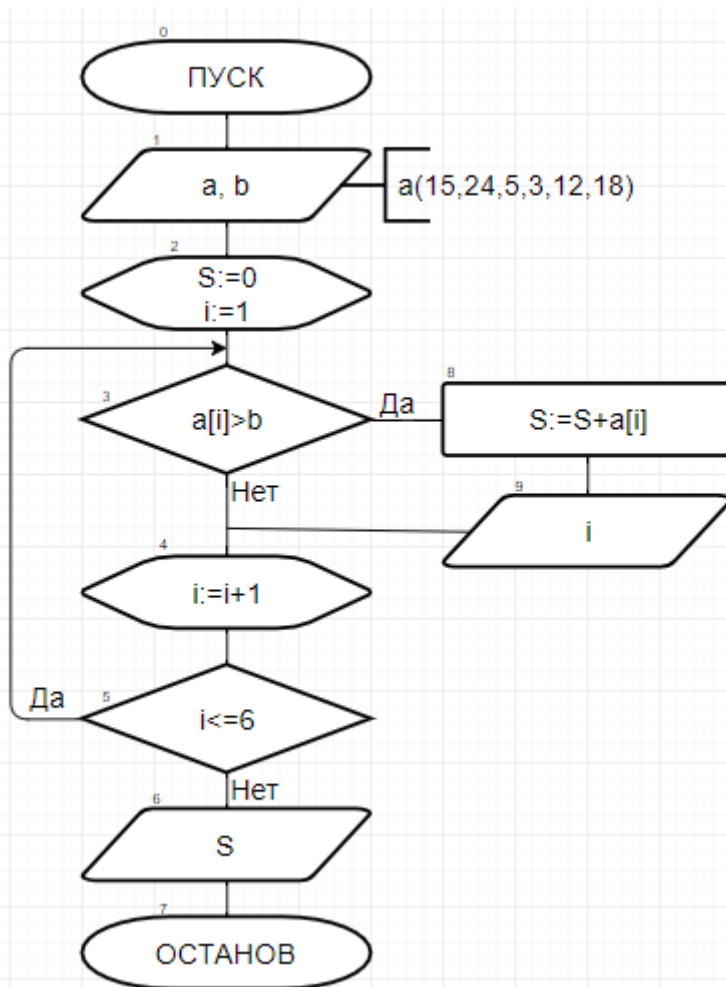
9. Результаты выполненной работы:



10. Анализ результатов вычисления: программа выводит сумму четных по значению элементов массива.

11. Вывод: при прохождении массива программа выбирает четные по значению элементы массива и складывает их.

1. Тема лабораторной работы: итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: изучение ИЦВП с управлением по индексу и функции, вариационного ряда с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: число вводится с клавиатуры. Нужно найти сумму всех элементов массива, значение которых больше числа, введенного с клавиатуры, и вывести индексы этих элементов.
5. Математическая модель:  
Если  $a[i] > b$ , где  $b$  введено с клавиатуры, то  $S := S + a[i]$ , вывести  $i$ .
6. Блок-схема:



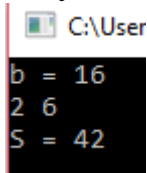
7. Список идентификаторов:

Имя	Тип	Смысл
a	integer	Массив с числами
S	integer	Сумма четных по значению элементов массива
i	integer	Индекс массива
b	integer	Введенное с клавиатуры число

8. Код программы:

```
program zadanie2;
const
  a: array [1..6] of integer = (15,24,5,3,12,18);
var
  S, i, b: integer;
begin
  write('b = ');
  readln(b);
  S:=0;
  for i:=1 to 6 do
    if a[i]>b then
      begin
        S:=S+a[i];
        write(i, ' ');
      end;
  writeln();
  writeln('S = ', S);
  readln();
end.
```

9. Результаты выполненной работы:



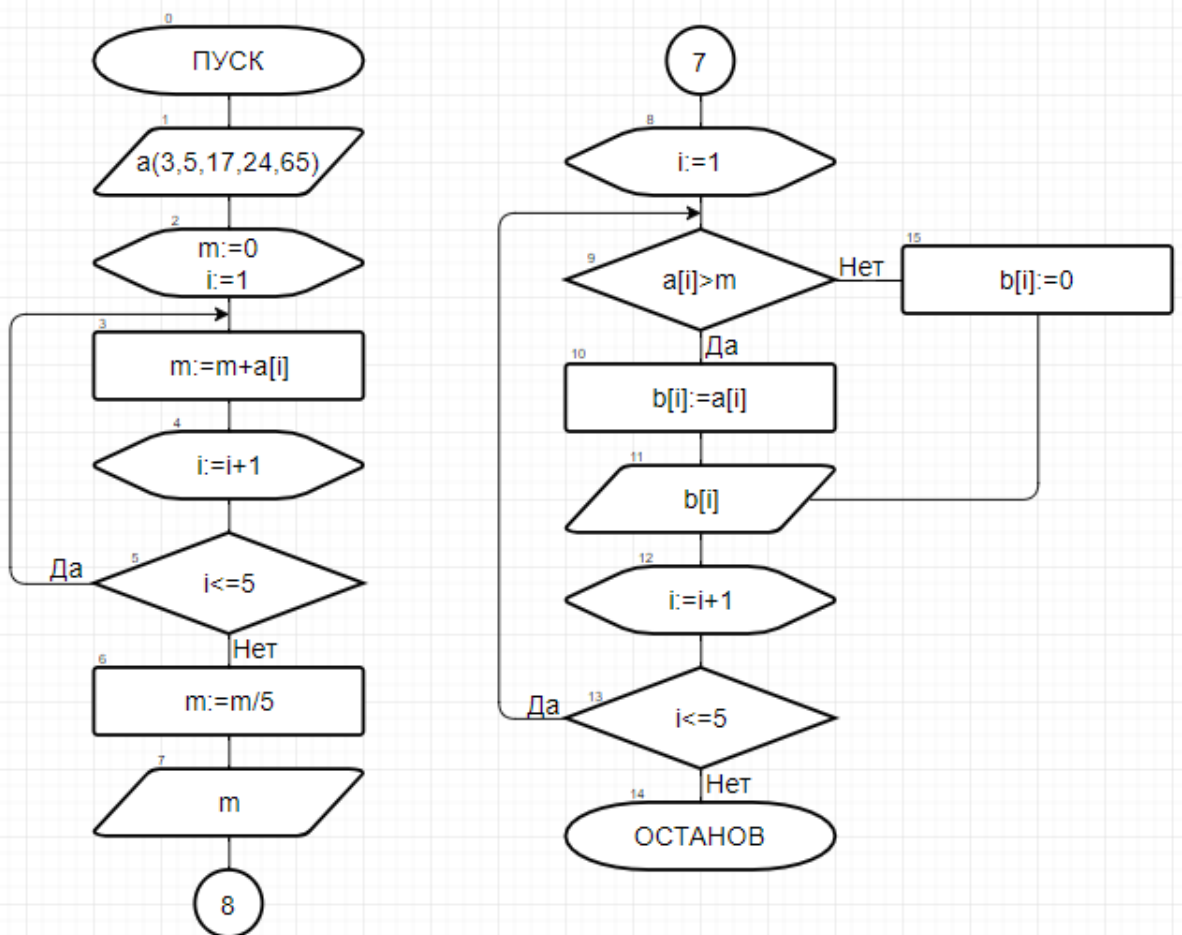
```
C:\User
b = 16
2 6
S = 42
```

10. Анализ результатов вычисления: программа выводит индексы

элементов массива, значение которых больше введенного числа, и сумму этих элементов.

11. Вывод: при прохождении массива сравниваются введенное число и значение элемента массива. Если элемент больше данного числа, то программа записывает значение в сумму и выводит номер элемента массива.

1. Тема лабораторной работы: итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: изучение ИЦВП с управлением по индексу и функции, вариационного ряда с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: найти среднее арифметическое массива, создать второй массив. Во втором массиве оставить числа первого массива, которые больше среднего арифметического, остальные заменить 0.
5. Математическая модель:  
Если элемент первого массива больше среднего арифметического, то  $b[i]=a[i]$ , иначе  $b[i]=0$ .
6. Блок-схема:



7. Список идентификаторов:

Имя	Тип	Смысл
a	integer	Первый массив
b	integer	Второй массив, состоящий из элементов первого, больших среднего арифметического
i	integer	Индекс массива
m	real	Сначала сумма элементов массива, потом среднее арифметическое

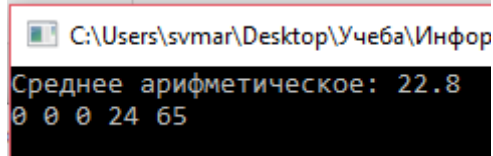
8. Код программы:

```

program zadanie3;
const
  a: array [1..5] of integer = (3,5,17,24,65);
var
  b: array [1..5] of integer;
  i: integer;
  m: real;
begin
  m:=0;
  for i:=1 to 5 do
    m:=m+a[i];
  m:=m/5;
  writeln('Среднее арифметическое: ', m:0:1);
  for i:=1 to 5 do
    begin
      if a[i]>m then
        b[i]:=a[i]
      else b[i]:=0;
      write(b[i], ' ');
    end;
  readln();
end.

```

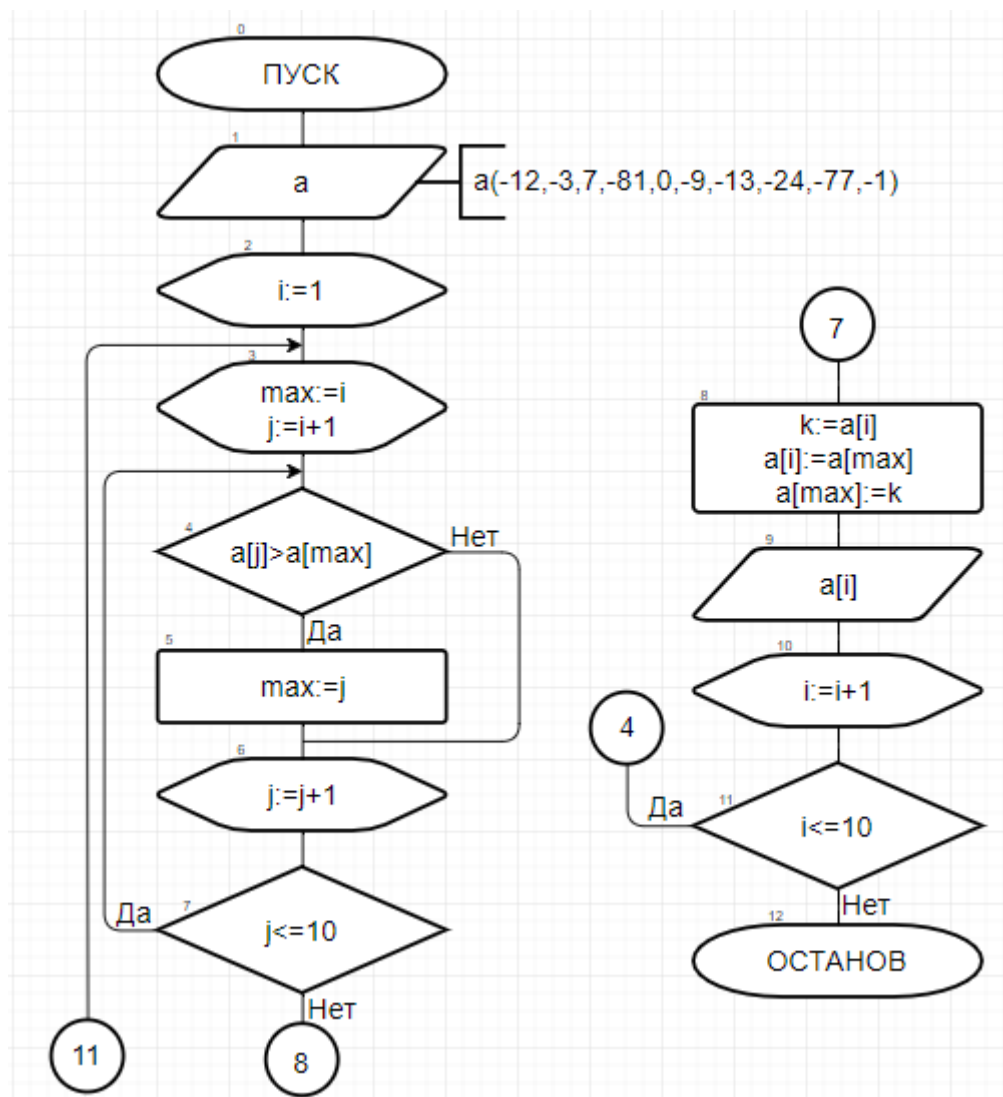
9. Результаты выполненной работы:



10. Анализ результатов вычисления: программа выводит среднее арифметическое элементов первого массива и второй массив.
11. Вывод: программа вычисляет среднее арифметическое первого массива, заполняет второй массив элементами первого, которые больше среднего арифметического этого же массива, остальные элементы равны 0.

1. Тема лабораторной работы: итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: изучение ИЦВП с управлением по индексу и функции, вариационного ряда с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: упорядочить одномерный массив по убыванию.
5. Математическая модель:  
Ищется максимальный элемент в массиве, меняется местами максимальный и первый элементы. Далее рассматривается массив без первого элемента, действия повторяются.

6. Блок-схема:



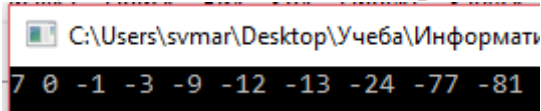
7. Список идентификаторов:

Имя	Тип	Смысл
a	integer	Массив
i	integer	Индекс массива для внешнего цикла
j	integer	Индекс массива для внутреннего цикла, где происходит поиск максимального значения
max	integer	Индекс максимального значения элемента массива

8. Код программы:

```
program zadanie4;
const
  a: array [1..10] of integer = (-12,-3,7,-81,0,-9,-13,-24,-77,-1);
var
  i, j, k, max, maxk: integer;
begin
  for i:=1 to 10 do
    begin
      max:=i;
      for j:=i+1 to 10 do
        if a[j]>a[max] then
          begin
            max:=j;
          end;
      k:=a[i];
      a[i]:=a[max];
      a[max]:=k;
      write(a[i], ' ');
    end;
    readln();
  end.
```

9. Результаты выполненной работы:



10. Анализ результатов вычисления: программа выводит отсортированный по убыванию массив.

11. Вывод: программа сортирует элементы массива по убыванию.

Происходит поиск максимального значения, который записывается в ячейку массива с текущим индексом, а прежнее значение записывается в ячейку максимального. Поиск максимально продолжается по элементам массива, исключая предыдущий максимальный элемент.