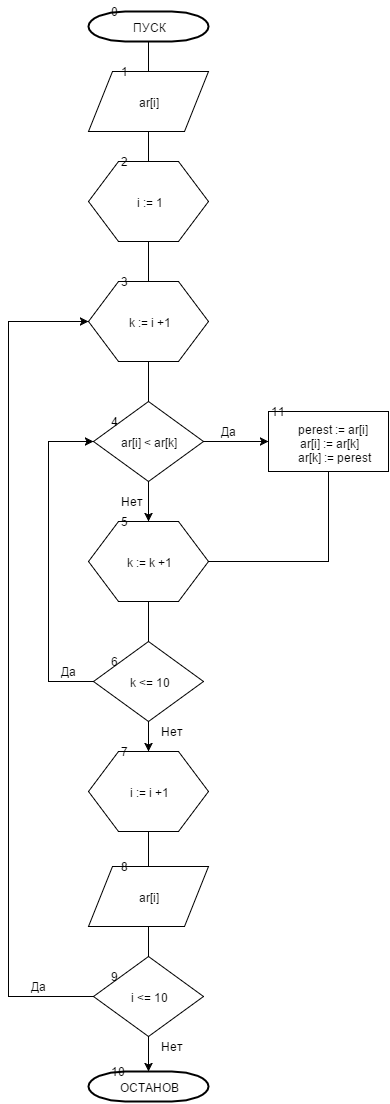
Лабораторная работа №10.

Задание 1.

1. Тема: Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: научиться строить вариационный ряд для решения задач средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Дан одномерный массив. Упорядочить его по убыванию.
5. Математическая модель:

Находим максимальное значение, сдвигаем его на самую «левую» позицию, в следующем этапе уже не учитываем его в проверке условия.

1. Блок-схема: см. ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| ar | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| k | integer | Параметр цикла, индекс |
| perest | integer | Для перестановки |

1. Код программы:

program lr10\_task1;

var

perest, i, k: integer;

ar: array [1..10] of integer;

begin

randomize;

for i := 1 to 10 do begin

ar[i] := random(50)-15;

write(ar[i],' ');

end;

writeln();

writeln();

writeln('Sortiruem po ubyvaniy: ');

writeln();

for I := 1 to 10 do begin

for k := i + 1 to 10 do begin

if ( ar[i] <ar[k] ) then begin

perest := ar[i];

ar[i] := ar[k];

ar[k] := perest;

end;

end;

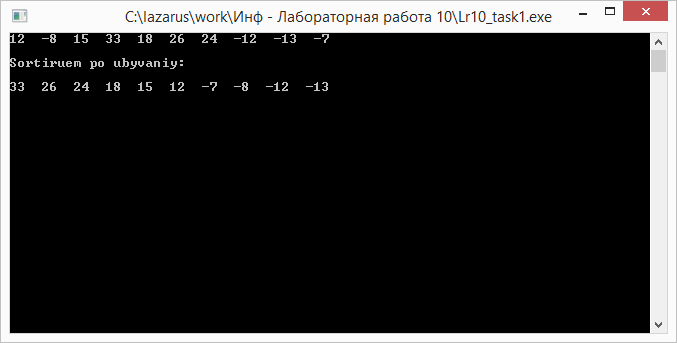
write(ar[i], ' ');

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Находим максимальное значение, сдвигаем его на самую «левую» позицию, в следующем этапе уже не учитываем его в проверке условия.

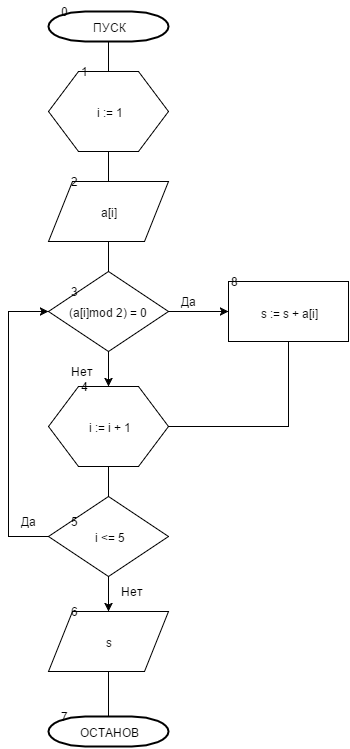
1. Вывод: программа работает корректно.

Задание 2.

1. Тема: Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: научиться строить вариационный ряд для решения задач средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Дан одномерный массив. Найти сумму четных (по значению) элементов массива.
5. Математическая модель:

Проверим значение каждого элемента на четность. Если четный, то прибавим значение к переменной суммы.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| s | integer | Сумма четных по значению элементов |

1. Код программы:

program Lr10\_task2;

var

a : array [1..5] of integer;

i,s : integer;

begin

for i := 1 to 5 do begin

readln(a[i]);

if (a[i] mod 2) = 0 then

s := s + a[i];

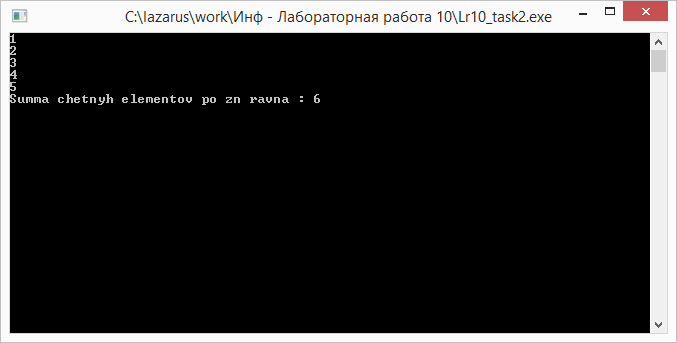
end;

writeln('Summa chetnyh elementov po zn ravna : ',s);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Проверяем элемент массива на четность и если он четный, прибавляем его значение к переменной суммы.

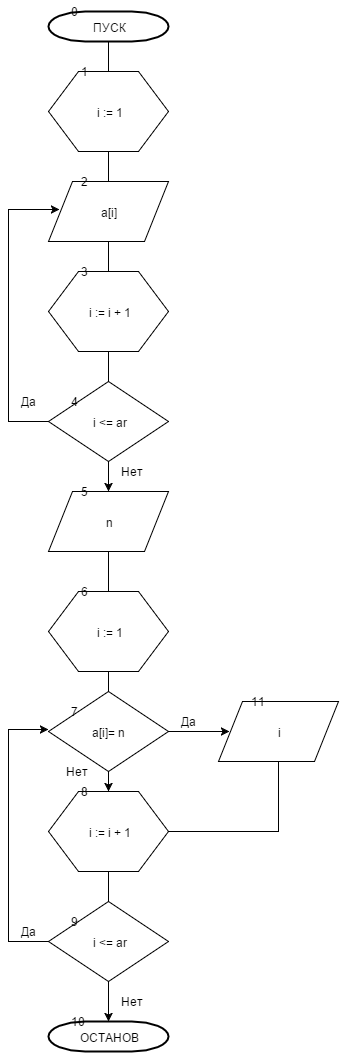
1. Вывод: программа работает корректно.

Задание 3.

1. Тема: Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: научиться строить вариационный ряд для решения задач средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Дан одномерный массив. С клавиатуры вводится число. Если такое число есть в массиве, то выводится его номер.
5. Математическая модель:

Проверим, равняется ли какой-нибудь элемент массива нашему числу. Если да, то выведем на экран его номер.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| n | integer | Число, номер которого мы щем в массиве |
| ar | integer | Кол-во элементов массива |

1. Код программы:

program Lr10\_task3;

var

a : array [1..5] of integer;

i,n,ar : integer;

begin

ar := 5;

for i := 1 to ar do begin

writeln('Vvedite ',i,' -i element massiva ');

readln(a[i]);

end;

writeln('Vvedite chislo dly proverki '); readln(n);

writeln();

for i:= 1 to ar do begin

if a[i] = n then

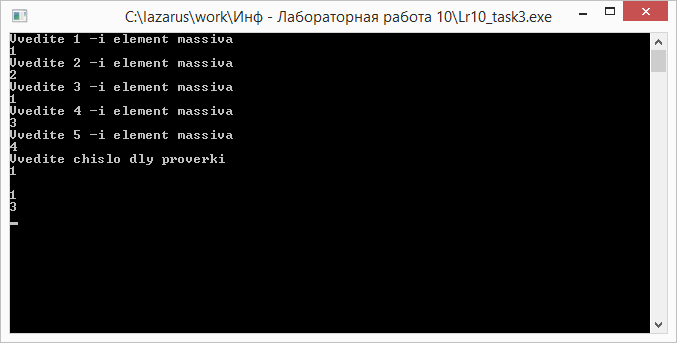
writeln(i);

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

В цикле проверим, равняется ли какой-нибудь элемент массива нашему числу. Если да, то выведем на экран его номер.

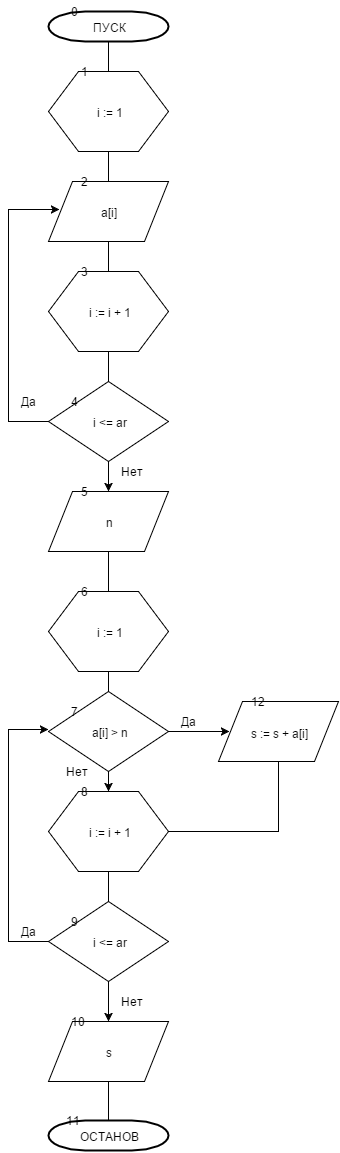
1. Вывод: программа работает корректно.

Задание 4.

1. Тема: Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: научиться строить вариационный ряд для решения задач средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Дан одномерный массив. С клавиатуры вводится число. Найти сумму всех элементов массива, значения которых больше числа, введенного с клавиатуры.
5. Математическая модель:

Если элемент массива больше заданного числа, то прибавялем его значение к переменной суммы.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| n | integer | Вводимое число |
| ar | integer | Кол-во элементов массива |
| s | integer | Сумма элеменов больше n |

1. Код программы:

program Lr10\_task4;

var

a : array [1..5] of integer;

i,n,ar,s : integer;

begin

ar := 5;

for i := 1 to ar do begin

writeln('Vvedite ',i,' -i element massiva ');

readln(a[i]);

end;

writeln('Vvedite chislo dly proverki '); readln(n);

for i:= 1 to ar do begin

if a[i] > n then

s := s + a[i];

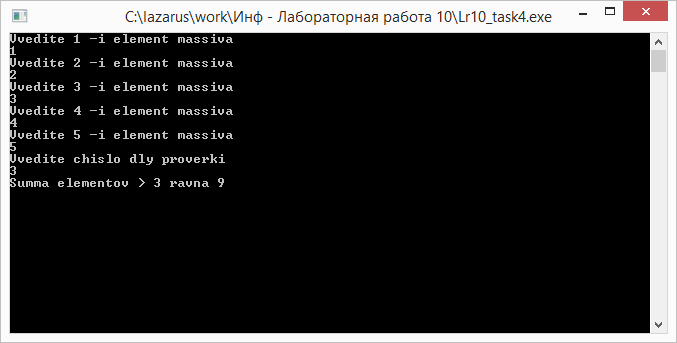
end;

writeln('Summa elementov > ',n,' ravna ',s);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

В цикле - если элемент массива больше заданного числа, то прибавялем его значение к переменной суммы.

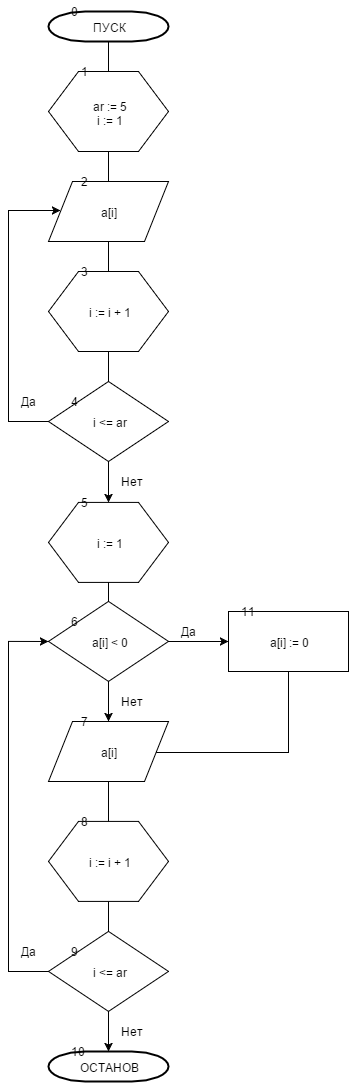
1. Вывод: программа работает корректно.

Задание 5.

1. Тема: Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: научиться строить вариационный ряд для решения задач средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: Дан одномерный массив. Заменить все отрицательные элементы нулями.
5. Математическая модель:

Если элемент массива отрицательный, то присваиваем ему значение 0.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| ar | integer | Кол-во элементов массива |

1. Код программы:

program Lr10\_task5;

var

a : array [1..5] of integer;

i,ar : integer;

begin

ar := 5;

for i := 1 to ar do begin

writeln('Vvedite ',i,' -i element massiva ');

readln(a[i]);

end;

for i:= 1 to ar do begin

if a[i] < 0 then

a[i] := 0;

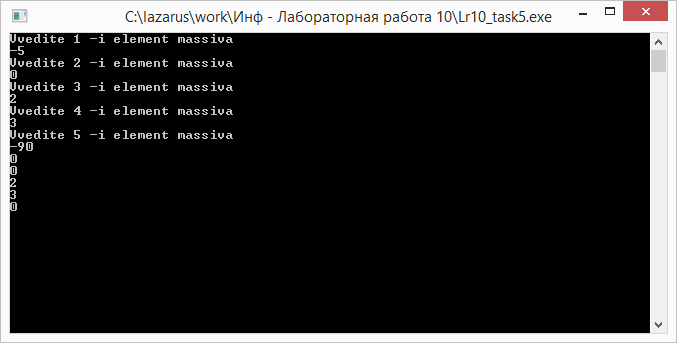
writeln(a[i]);

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

В цикле - если элемент массива отрицательный, то присваиваем ему значение 0.

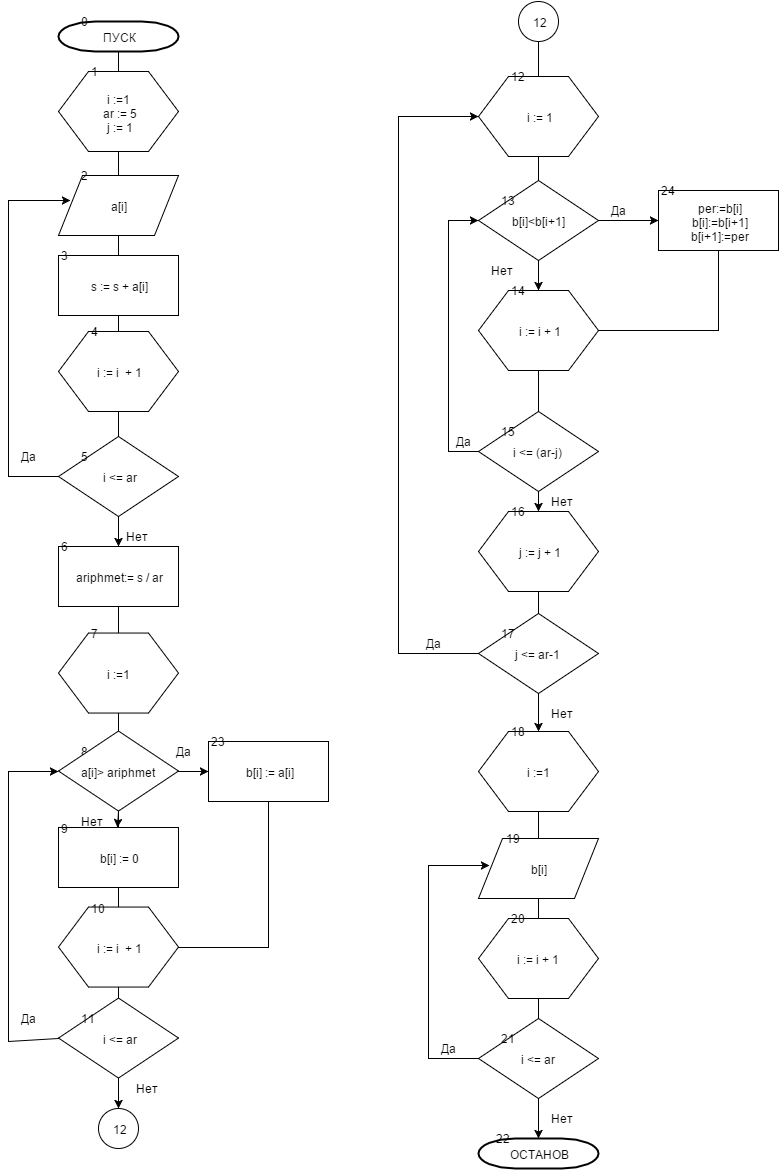
1. Вывод: программа работает корректно.

Задание 6.

1. Тема: Итерационные ЦВП с управлением по индексу и функции. Вариационный ряд.
2. Цель: научиться строить вариационный ряд для решения задач средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: дан одномерный массив. Найти его среднее арифметическое. Составить второй массив, элементами которого будут элементы первого массива, которые больше среднего арифметического. Остальные элементы заменить нулями. Массив отсортировать по убыванию.
5. Математическая модель:

Считаем среднее арифметическое. Проверяем какие из элементов массива больше ср. арифм. и присваиваем их значения элементам второго массива. Затем отсортировываем получившийся массив по убыванию.

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Array of integer | Числовой массив |
| b | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| ar | integer | Кол-во элементов массива |
| s | integer | Сумма |
| j | integer | Параметр цикла |
| per | integer | Для перестановки |
| ariphmet | real | Среднее арифметическое |

program Lr10\_task6;

var

a : array [1..5] of integer;

b : array [1..5] of integer;

i,ar,s,j,per : integer;

ariphmet : real;

begin

ar := 5;

for i := 1 to ar do begin

writeln('Vvedite ',i,' -i element massiva ');

readln(a[i]);

s := s + a[i];

end;

ariphmet := s / ar;

for i := 1 to ar do

if a[i] > ariphmet then

b[i] := a[i] else

b[i] := 0;

for j:=1 to ar-1 do

for i:=1 to ar-j do

begin

If b[i] < b[i+1] then

begin

per := b[i];

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

b[i+1] := per;

end;

end;

writeln('Srednee ariphemticheskoe elementov massiva A : ',ariphmet:2:5);

writeln;

writeln('Otsortirovanny po ubyvaniu massiv B:');

for i := 1 to ar do begin

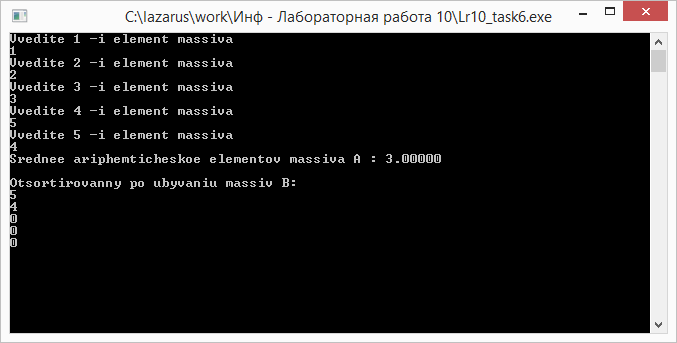
writeln(b[i]);

end;

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления:

Считаем среднее арифметическое. Проверяем какие из элементов массива больше ср. арифм. и присваиваем их значения элементам второго массива. Затем отсортировываем получившийся массив по убыванию.

Вывод: программа работает корректно.