Лабораторная работа №10.

**Тема: Многоступенчатые циклические вычислительные процессы.**

**Двумерные массивы.**

**Цель:**

**Используемое оборудование: ПК, среда программирования Lazarus.**

**Задание 1.**

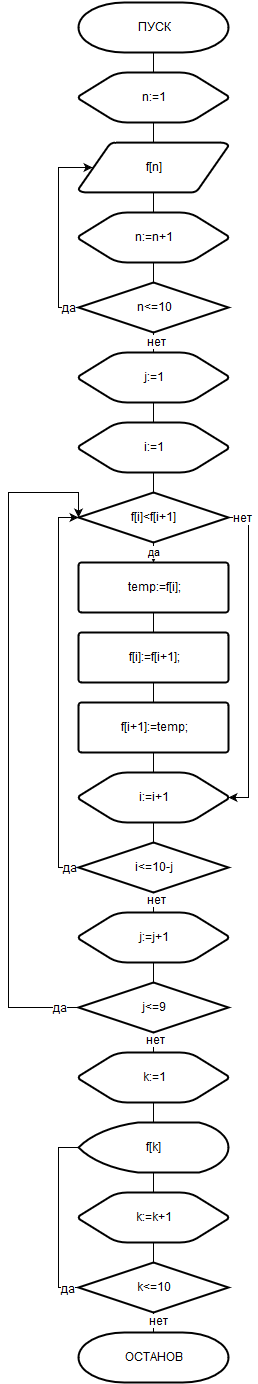
1. Постановка задачи: Отсортировать массив по убыванию
2. Математическая модель:

temp:=f[i];

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

f[i+1]:=temp;

1. Блок-схема



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| j | integer | Параметр цикла |
| k | integer | Параметр цикла |
| f | array of integer | Массив |
| temp | integer | Временная переменная |
| n | integer | Параметр цикла |

1. **Код программы:**

program zad1;

var

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

i,j,k,temp,n:integer;

begin

for n:=1 to 10 do

begin

writeln('vvedite ',n,'-i element massiva');

readln(f[n]);

end;

for j:=1 to 9 do

for i:=1 to 10-j do

begin

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

begin

temp:=f[i];

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

f[i+1]:=temp;

end;

end;

write('[');

for k:=1 to 10 do

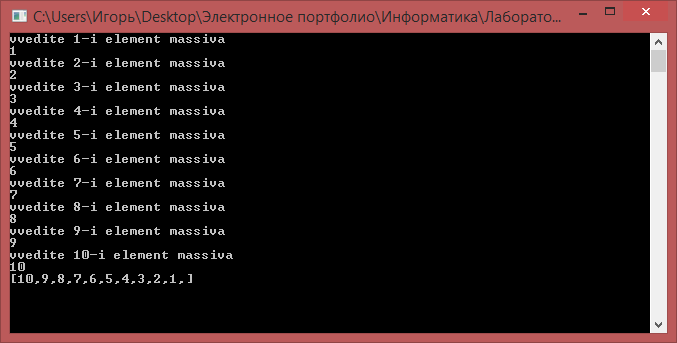
write(f[k],',');

write(']');

readln()

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа сортирует массив по убыванию
2. Вывод: Программа работает корректно.

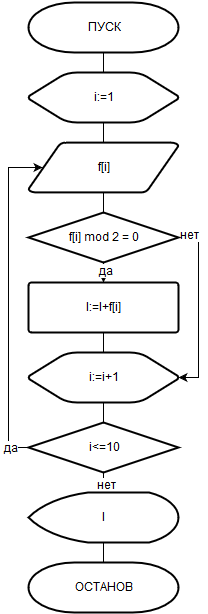
**Задание 2.**

1. Постановка задачи: Найти сумму четных по значению элементов массива
2. Математическая модель:

if f[i] mod 2 = 0 then

l:=l+f[i

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| j | integer | Параметр цикла |
| f | array of integer | Mассив |

1. **Код программы:**

program zad2;

var

i,l:integer;

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

begin

l:=0;

for i:=1 to 10 do

begin

writeln('vvedite ',i,'-tyi element massiva');

readln(f[i]);

if f[i] mod 2 = 0 then

l:=l+f[i]

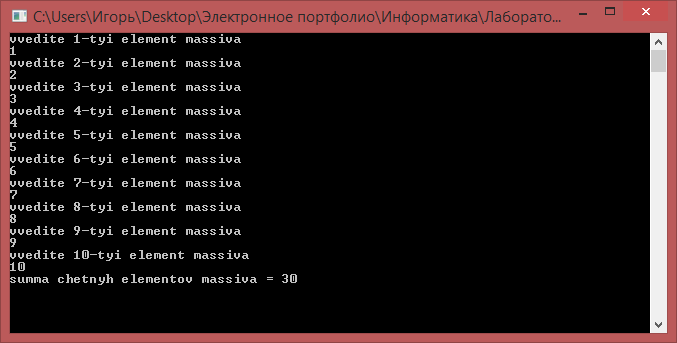
end;

writeln('summa chetnyh elementov massiva = ',l);

readln();

end.

1. Результат выполнения программы:



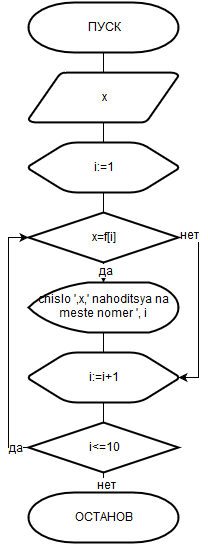
1. Анализ результатов вычисления: Программа вычисляет сумму четных по зачению элементов массива
2. Вывод: Программа работает корректно.

**Задание 3.**

1. Постановка задачи: Вывести номер элемента в массиве.
2. Математическая модель:

x=f[i]

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| x | Integer | Число |
| f | array of integer | Массив |

1. **Код программы:**

program zad3;

const

f:array [1..10] of integer = (11,-4,-12,13,5,7,3,-5,-10,13);

var

x,i:integer;

begin

writeln('vvedite chislo');

readln(x);

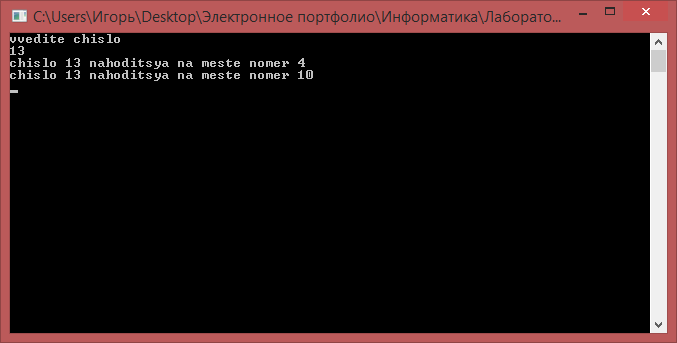
for i :=1 to 10 do

If x=f[i] then writeln('chislo ',x,' nahoditsya na meste nomer ', i);

readln()

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа выводит номер элемента в массиве
2. Вывод: Программа работает корректно

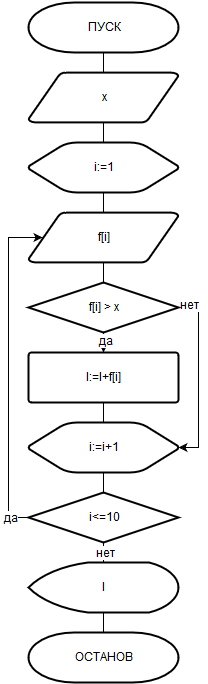
**Задание 4.**

1. Постановка задачи: Найти сумму тех элементов массива, которые больше заданного числа
2. Математическая модель:

if f[i] > m then

l:=l+f[i];

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | Integer | Параметр цикла |
| l | Integer | Сумма |
| x | Integer | Вводимое число |
| f | array of integer | Массив |

1. **Код программы:**

program zad4;

var

i,l,x:integer;

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

begin

l:=0;

writeln('vvedite chislo');

readln(x);

for i:=1 to 10 do

begin

writeln('vvedite ',i,' element massiva');

readln(f[i]);

if f[i] > x then

l:=l+f[i];

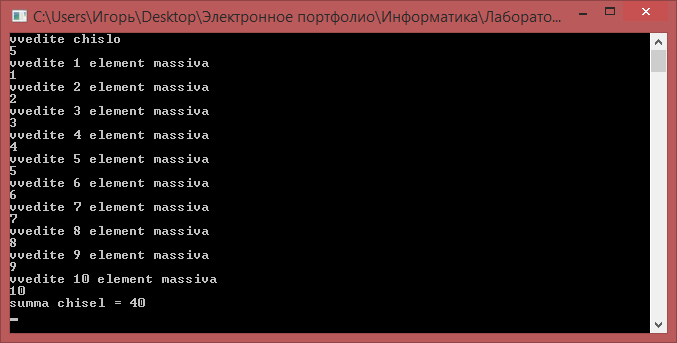
end;

writeln('summa chisel = ', l);

readln();

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа считает сумму тех элементов массива, которыые больше заданного числа.
2. Вывод: Программа работает корректно.

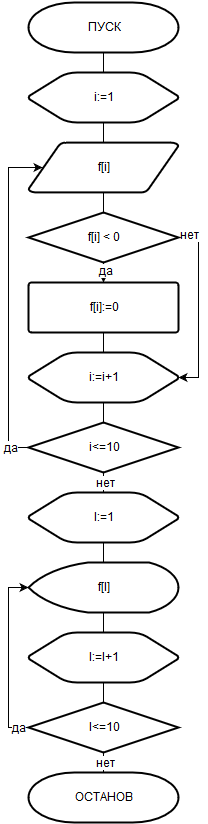
**Задание 5.**

1. Постановка задачи: Заменить отрицательные элементы массива нулями.
2. Математическая модель:

if f[i] < 0 then

f[i]:=0;

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| l | integer | Параметр цикла |
| f | array of integer | Массив |

1. **Код программы:**

program zad5;

var

i, l:integer;

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

begin

l:=0;

for i:=1 to 10 do

begin

writeln('vvedite ',i,' element massiva');

readln(f[i]);

if f[i] < 0 then

f[i]:=0;

end;

write('[');

for l:=1 to 10 do

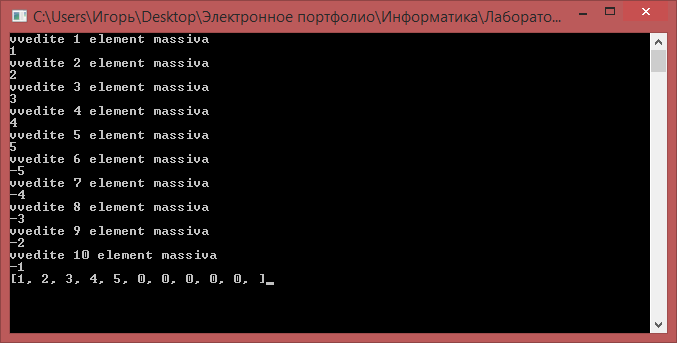
write(f[l],', ');

write(']');

readln();

end.

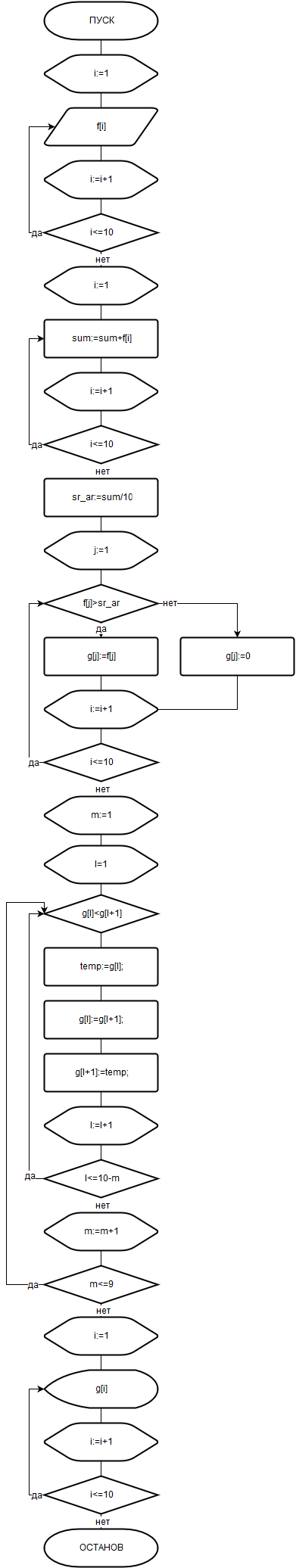
1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа заменяет отрицательные элементы массивва нулями
2. Вывод: программа работает корректно

**Задание 6.**

1. 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.
2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | Integer | Параметр цикла |
| j | Integer | Параметр цикла |
| sum | integer | Сумма |
| temp | integer | Временная переменная |
| m | integer | Параметр цикла |
| l | integer | Параметр цикла |
| sr\_ar | real | среднее арифметическое |
| f | array of integer | Первоначальный массив |
| g | array of integer | Результирующий массив |

1. **Код программы:**

program zad6;

var

f,g:array [1..10] of integer;

i,j,sum,m,l,temp:integer;

sr\_ar:real;

begin

sum:=0;

for i:=1 to 10 do

begin

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

readln(f[i]);

end;

for i:=1 to 10 do

sum:=sum+f[i];

sr\_ar:=sum/10;

for j:=1 to 10 do

begin

if f[j]>sr\_ar then

g[j]:=f[j]

else

g[j]:=0;

end;

for m:=1 to 9 do

for l:=1 to 10-m do

if g[l]<g[l+1] then

begin

temp:=g[l];

g[l]:=g[l+1];

g[l+1]:=temp;

end;

i:=0;

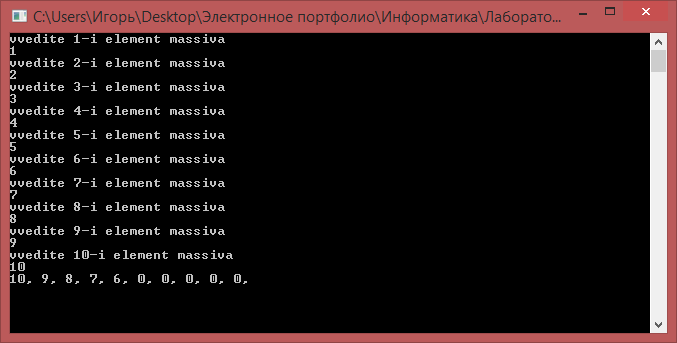
for i:=1 to 10 do

write(g[i],', ');

readln;

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа находит среднее арифметическое элементов массива. Элементы, которые больше среднего арифметического, заносит во второй массив, а вместо других заносит нули. После чего сортирует второй массив по убыванию.
2. Вывод: программа работает корректно