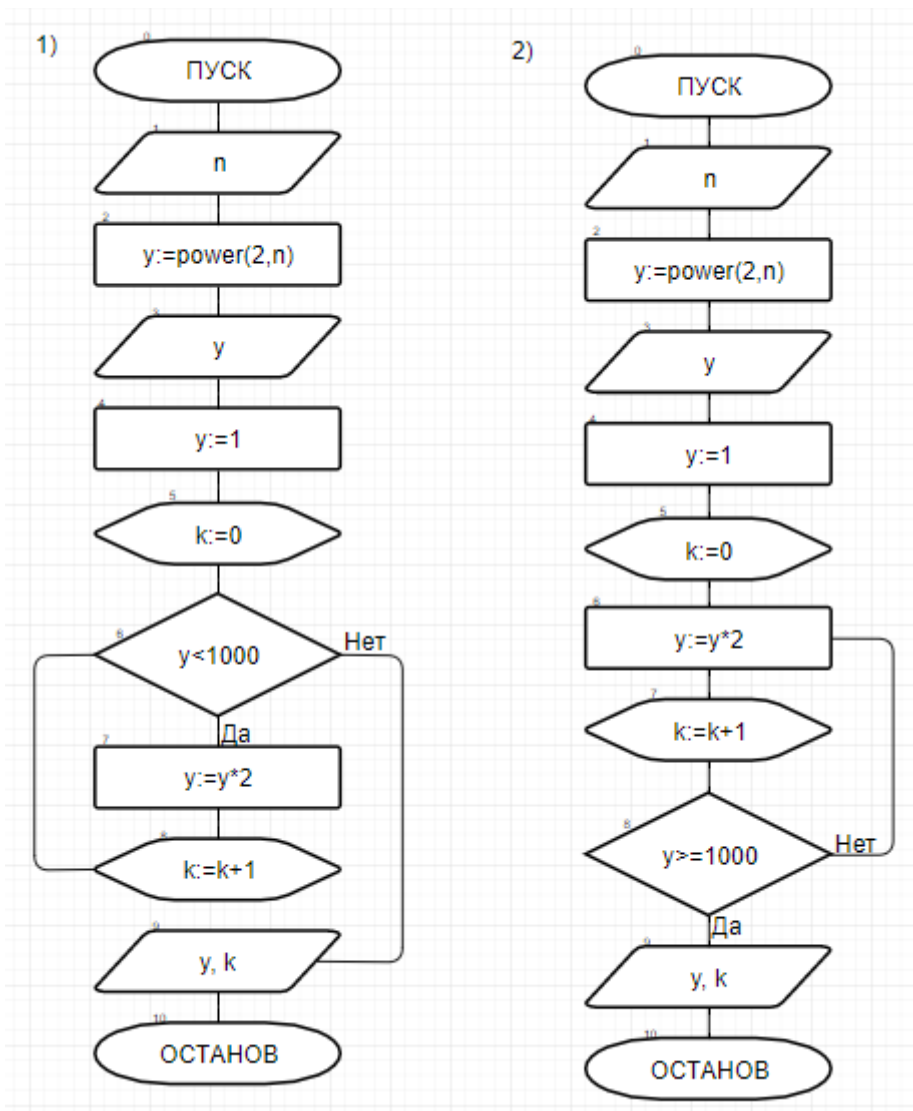


Лабораторная работа № 7

1. Тема лабораторной работы: итерационные циклические вычислительные процессы с управлением по функции.
2. Цель: изучение итерационных вычислительных процессов с управлением по функции с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: вычислить 2^n и определить первое значение степени, при котором $2^n \geq 1000$, используя 1) цикл с предусловием, 2) цикл с постусловием.
5. Математическая модель: $y = 2^n$
6. Блок-схема:



7. Список идентификаторов:

Имя	Тип	Смысл
y	real	Значение 2^n
n	integer	Степень 2
k	integer	Параметр цикла

8. Код программы:

1) program zadanie11;

uses

Math;

var

y: real;

n, k: integer;

begin

write('Введите n ');

readln(n);

y:=power(2,n);

writeln('2^n = ', y:1:0);

y:=1;

k:=0;

while y<1000 do

begin

y:=y*2;

k:=k+1;

end;

writeln('Значение больше 1000 2^n = ', y:1:0);

writeln('n = ', k);

readln();

end.

2) program zadanie1;

uses

Math;

var

y: real;

n, k: integer;

begin

write('Введите n ');

readln(n);

y:=power(2,n);

writeln('2^n = ', y:1:0);

y:=1;

k:=0;

repeat

begin

y:=y*2;

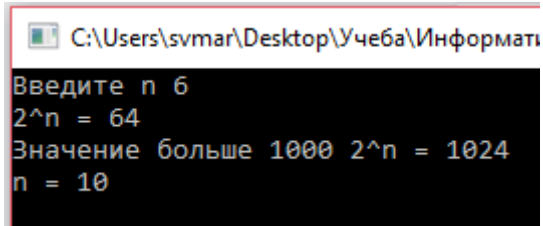
k:=k+1;

end;

until y>=1000;

```
writeln('Значение больше 1000 2^n = ', y:1:0);  
writeln('n = ', k);  
readln();  
end.
```

9. Результаты выполненной работы:

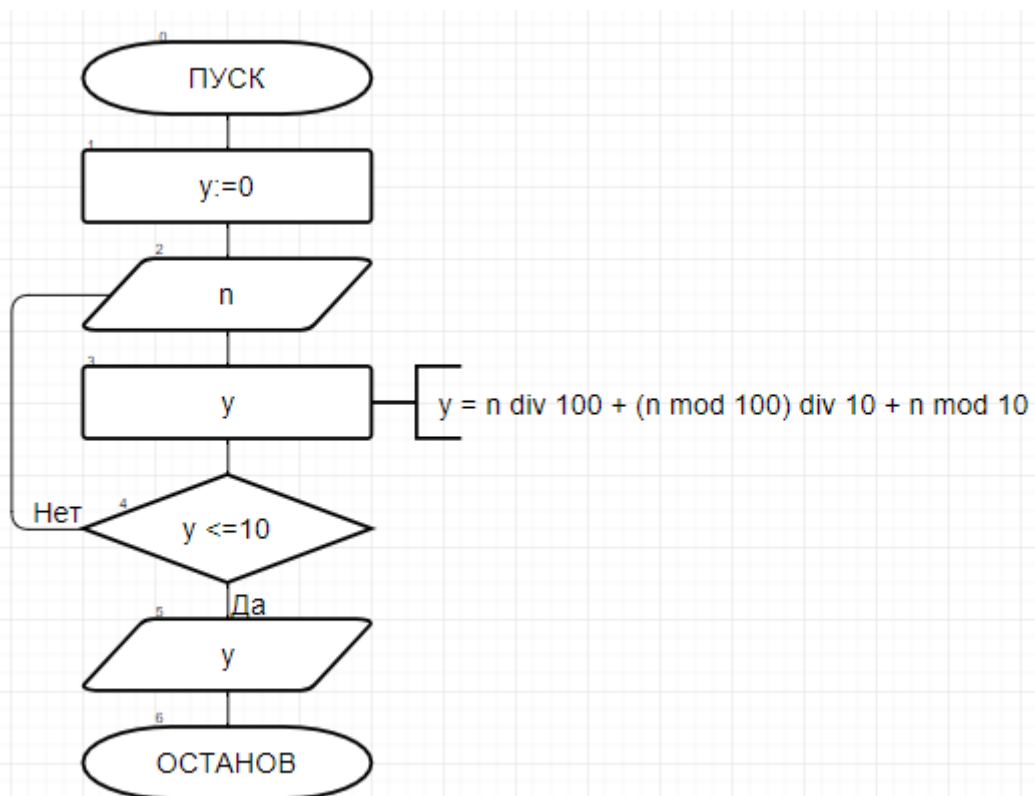


```
C:\Users\svmar\Desktop\Учеба\Информати  
Введите n 6  
2^n = 64  
Значение больше 1000 2^n = 1024  
n = 10
```

10. Анализ результатов вычисления: в обоих вариантах программа выводит 2^n , где n вводится с клавиатуры и n , при котором значение $2^n > 1000$.
11. Вывод: программа считает 2^n , где n вводится с клавиатуры. А степень при которой значение больше 1000 считается двумя различными циклами, с предусловием и постусловием, изменяя само значение функции (умножение на 2), а также степень числа (увеличением на 1).

1. Тема лабораторной работы: итерационные циклические вычислительные процессы с управлением по функции.
2. Цель: изучение итерационных вычислительных процессов с управлением по функции с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: вычислить сумму цифр трехзначного числа, вводимого с клавиатуры, если полученный результат больше 10, то вводится следующее трехзначное число, иначе программа завершается.
5. Математическая модель:

$$y = n \operatorname{div} 100 + (n \operatorname{mod} 100) \operatorname{div} 10 + n \operatorname{mod} 10$$
6. Блок-схема:



7. Список идентификаторов:

Имя	Тип	Смысл
y	integer	Сумма цифр вводимого числа
n	integer	Вводимое число

8. Код программы:

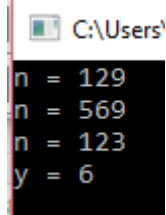
```

program zadanie2;
var
  n, y: integer;
begin
  y:=0;
  repeat
    begin

```

```
write('n = ');  
readln(n);  
y:=n div 100 + (n mod 100) div 10 + n mod 10;  
end;  
until y<=10;  
writeln('y = ', y);  
readln();  
end.
```

9. Результаты выполненной работы:



```
C:\Users'  
n = 129  
n = 569  
n = 123  
y = 6
```

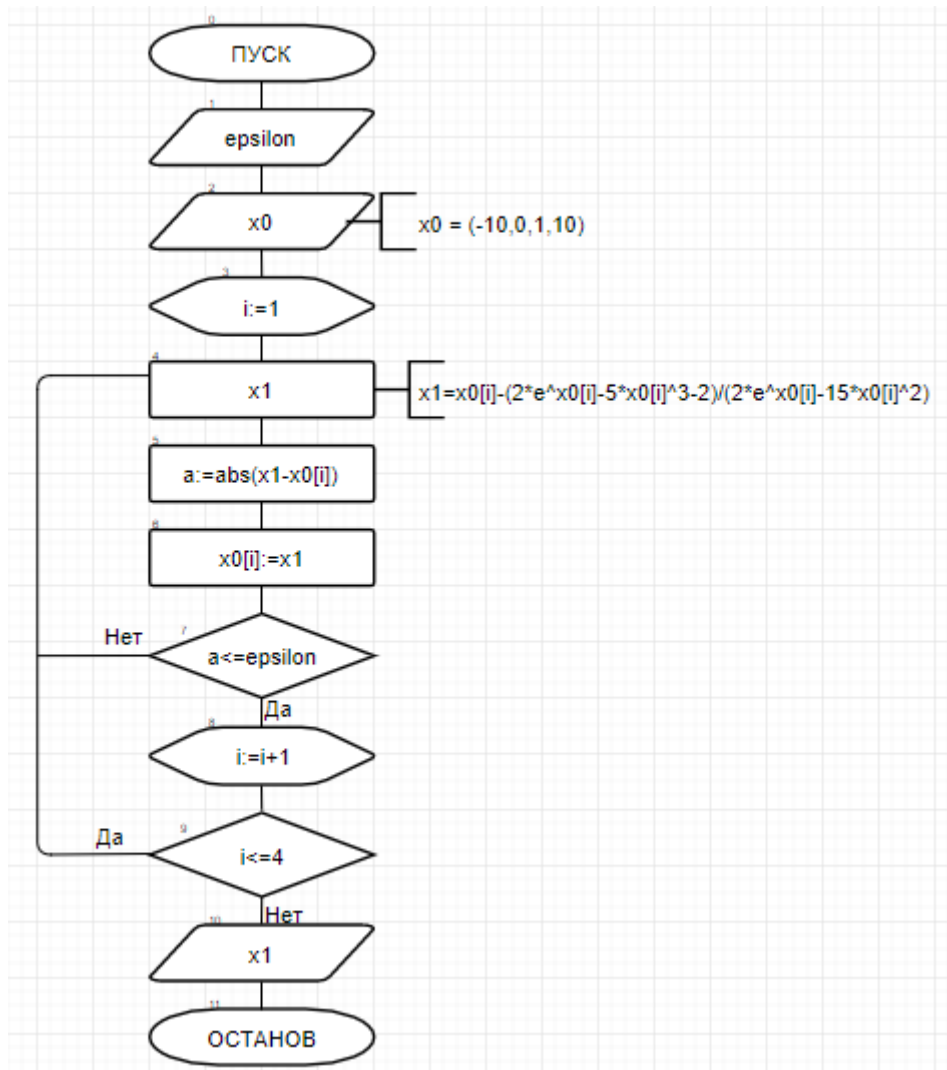
10. Анализ результатов вычисления: пользователь вводит трехзначные числа до того момента, пока сумма цифр последнего введенного числа будет не меньше 10. После этого программа выводит сумму цифр последнего числа.
11. Вывод: программа вычисляет сумму цифр вводимых пользователем трехзначных чисел до того момента пока сумма цифр не будет меньше 10, после этого программа завершается выводом суммы цифр последнего введенного числа.

1. Тема лабораторной работы: итерационные циклические вычислительные процессы с управлением по функции.
2. Цель: изучение итерационных вычислительных процессов с управлением по функции с помощью среды программирования Lazarus на языке Pascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: решить нелинейное уравнение методом Ньютона.
5. Математическая модель:

$$2 * e^x = 5 * x^3 + 2, [-10,10], \text{ с точностью } 10^{-6}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

6. Блок-схема:



7. Список идентификаторов:

Имя	Тип	Смысл
epsilon	real	Точность вычисления
x0	real	Массив с начальными значениями отрезка
x1	real	Более точный корень, чем x0[i]

a	real	Разность x0 и x1 для сравнения с точностью
i	integer	Параметр массива

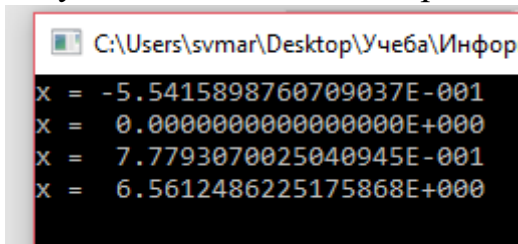
8. Код программы:

```

program zadanie3;
uses
  Math;
const
  epsilon=0.000001;
  x0: array[1..4] of real = (-10,0,1,10);
var
  x1, a: real;
  i: integer;
begin
  for i:=1 to 4 do
  begin
    repeat
    begin
      x1:=x0[i]-(2*exp(x0[i])-5*(x0[i])*(x0[i])*(x0[i])-2)/(2*exp(x0[i])-
15*(x0[i])*(x0[i]));
      a:=abs(x1-x0[i]);
      x0[i]:=x1;
    end;
    until a<=epsilon;
    writeln('x = ',x1);
  end;
  readln();
end.

```

9. Результаты выполненной работы:



```

C:\Users\svmar\Desktop\Учеба\Информ
x = -5.5415898760709037E-001
x = 0.0000000000000000E+000
x = 7.7793070025040945E-001
x = 6.5612486225175868E+000

```

10. Анализ результатов вычисления: программа выводит корни данного уравнения.

11. Вывод: программа считает корни данного уравнения методом Ньютона, выводимые значение является наиболее точными значениями корней с заданной точностью. Для поиска корней используется метод Ньютона основанный на поиске корня при помощи касательных к графику функции. Массив задает отрезки проверки наличия корней.