**Лабораторная работа № 7**

**Тема:** Итерационные циклические вычислительные процессы с управлением по функции

**Цель:** Научиться реализовывать алгоритм итерационных вычислительных процессов с управлением по функции средством при Pascal

**Оборудование:**

* ПК
* Среда программирования Lazarus
* Интернет

**Задание 1**

**Постановка задачи**

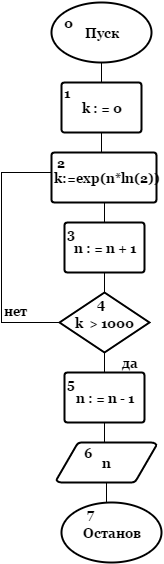
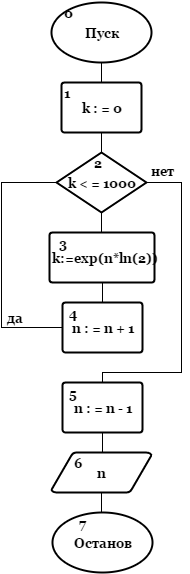
Вычислить 2 в степени n и при этом определить первое значение степени, при котором результат будет превышать значение 1000.

**Математическая модель**

k:=exp(n\*ln(2));

**Блок – схема**

***цикл с предусловием   цикл с постусловием***

****

**Список идентификаторов**

|  |  |  |
| --- | --- | --- |
| **имя** | **тип** | **смысл** |
| **n** | **integer** | **число** |
| **k** | **real** | **результат** |

**Код программы**

***цикл с предусловием***

program z1;

var

n:integer;

k:real;

begin

k:=0;

while k<=1000 do

begin

k:=exp(n\*ln(2));

n:=n+1;

end;

n:=n-1;

writeln('2 v ',n,' ','> 1000');

readln;

end.

***цикл с постусловием***

program lab7\_z1\_2;

var

n:integer;

k:real;

begin

k:=0;

repeat

k:=exp(n\*ln(2));

n:=n+1;

until k>=1000 ;

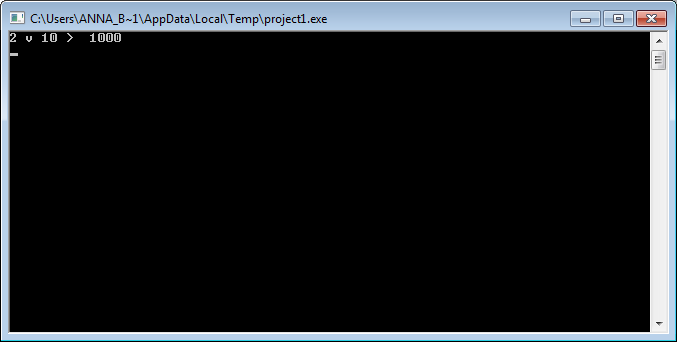
n:=n-1;

writeln('2 v ',n,' ','> 1000');

readln;

end.

**Результат выполнения программы**

****

**Анализ**

Программа выводит на экран первую степень двойки, превышающую число 1000.

**Задание 2**

**Постановка задачи**

**Индивидуальная задача**

Решить нелинейное уравнение методом Ньютона. Файл с индивидуальными задачами находится внизу страницы в ПРИЛОЖЕНИИ

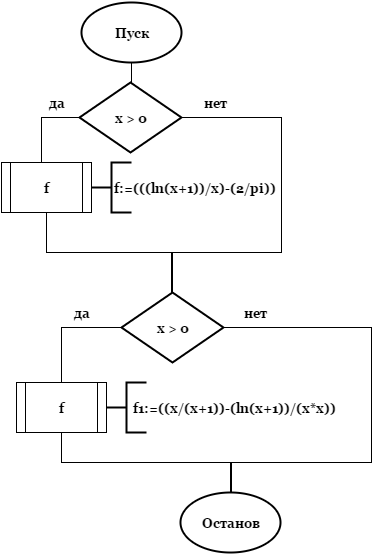
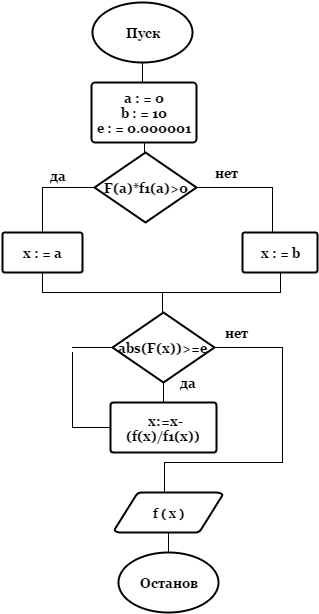
***Ln ( 1 + x ) / x = 2 / π***

На отрезке от 0 до 10 с точностью 10-6

**Математическая модель**

*Ln ( 1 + x ) / x = 2 / π*

**Блок – схема**

****

**Список идентификаторов**

|  |  |  |
| --- | --- | --- |
| **имя** | **тип** | **смысл** |
| f | real | функция |
| f1 | real | производная |
| a | integer | 0 |
| b | integer | 10 |
| e | real | точность 10-6 |
| x | real | значение |

**Код программы**

program lab7\_z2;

function f(x:real):real;

begin

if x > 0 then f:=(((ln(x+1))/x)-(2/pi));

end;

function f1(x:real):real;

begin

if x >0 then f1:=((x/(x+1))-(ln(x+1))/(x\*x));

end;

var

x,e:real;

a,b:integer;

begin

a:=0;

b:=10;

e:=0.000001;

if (F(a)\*f1(a)>0) then x:=a

else x:=b;

while abs(F(x))>=e do

begin

x:=x-(f(x)/f1(x));

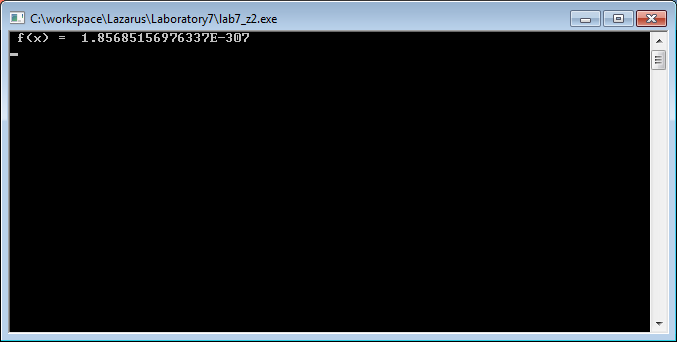
end;

writeln(' f(x) = ',F(x));

readln;

end.

**Результат выполнения программы**

****

**Анализ**

Программа выводит на экран вычисление нелинейного уравнения методом Ньютона.

**Задание 3**

**Постановка задачи**

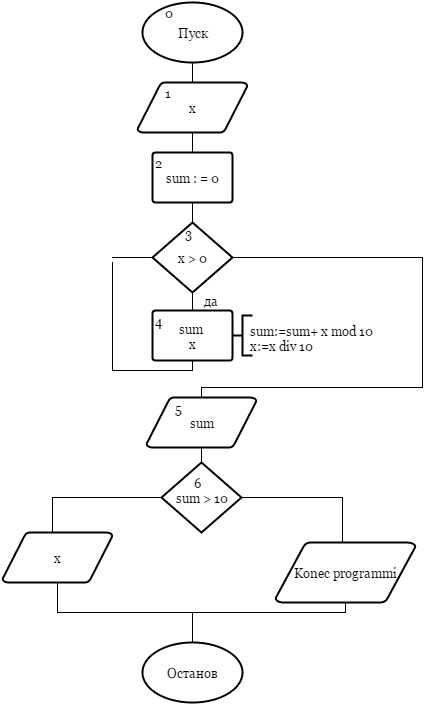
С клавиатуры вводится трехзначное число, считается сумма его цифр. Если сумма цифр числа больше 10, то вводится следующее трехзначное число, если сумма меньше либо равна 10 – программа завершается.

**Математическая модель**

sum:=sum+ x mod 10;

x:=x div 10;

**Блок – схема**

****

**Список идентификаторов**

|  |  |  |
| --- | --- | --- |
| **имя** | **тип** | **смысл** |
| x | integer | трёхзначное число |
| sum | integer | сумма цифр числа |

**Код программы**

program z3;

var

sum,x:integer;

begin

write ('Vvedite trehznachnoe chislo - ');

readln(x);

sum:=0;

while x>0 do

begin

sum:=sum+ x mod 10;

x:=x div 10;

end;

writeln('summa = ', sum);

if sum>10 then

begin

write ('Vvedite trehznachnoe chislo - ');

readln;

end

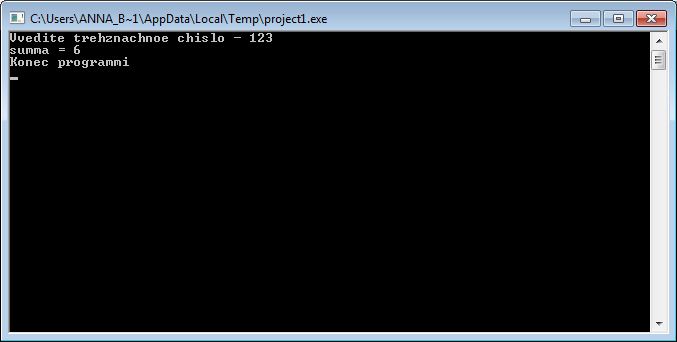
else

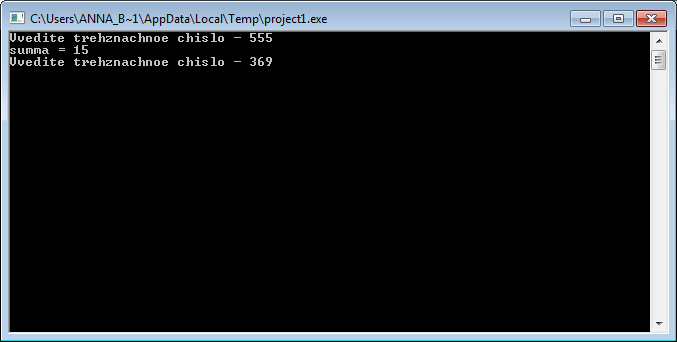
writeln('Konec programmi');

readln;

end.

**Результат выполнения программы**

****

****

**Анализ**

Программа выводит на экран сумму трёхзначного числа и если она больше 10, необходимо ввести следующее трёхзначное число, если меньше, то программа заканчивается.

**Задание 4**

**Постановка задачи**

Составить программу подсчета суммы факториалов целых чисел, где сумма не превышает число А, которое вводится с клавиатуры. На экран вывести сумму и все слагаемые.

**Математическая модель**

f:=1;

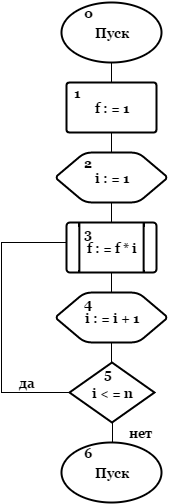
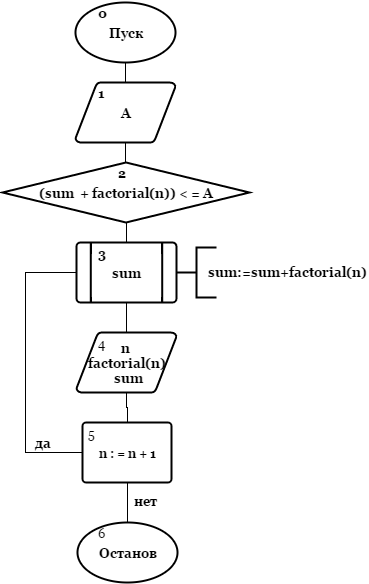
for i:=1 to n do

f:=f\*i;

sum:=sum+factorial(n);

n:=n+1;

**Блок – схема**

****

**Список идентификаторов**

|  |  |  |
| --- | --- | --- |
| имя | тип | смысл |
| i | integer | параметр цикла |
| sum | integer | сумма факториалов |
| A | integer | число, с которым сравнивается сумма факториалов |
| factorial | integer | функция вычисляющая факториал |
| n | integer | число |
| f | integer | вспомогательная переменная |

**Код программы**

program z4;

function factorial(n:integer):integer;

var

i,f:integer;

begin

f:=1;

for i:=1 to n do

f:=f\*i;

end;

var

A,sum,n:integer;

begin

writeln('Vvedite chislo A - ');

readln(A);

while (sum+factorial(n)) <=A do

begin

sum:=sum+factorial(n);

writeln(n,'! = ',factorial(n),' ','Summa = ',sum );

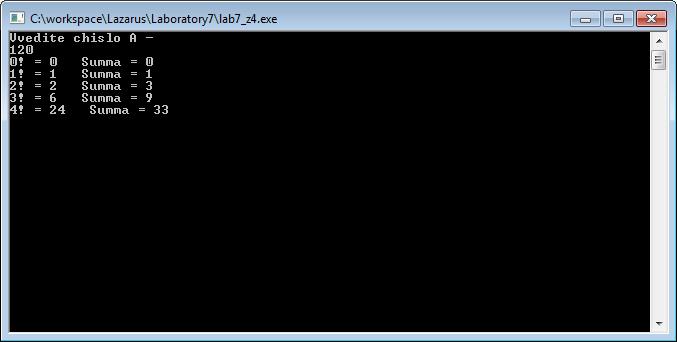
n:=n+1;

end;

readln;

end.

**Результат выполнения программы**

****

**Анализ**

Программа выводит на экран сумму факториалов и все слагаемые целых чисел, если сумма не превышает введённое число А.