**Лабораторная работа № 8 .**

Тема: итерационные циклические вычислительные процессы с управлением по индексу/аргументу и функции

Цель: научиться реализовывать итерационные циклические вычислительные процессы с управлением по индексу/аргументу и функции средствами FreePascal.

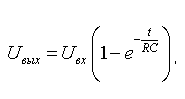
Используемое оборудование: ПК, среда программирования Lazarus, сайт draw.io

*Далее к каждому заданию будет начинаться с пункта №4(постановка задачи).*

**Задание 1.**

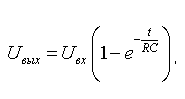
**Постановка задачи:**

Дан процесс, связанный с изменением выходного напряжения *Uвых*  на обкладках конденсатора электрической цепи, которая включает активное сопротивление *R* = 2 Ом и конденсатор с емкостью С=0.01 Ф.  Построить переходную характеристику заряда конденсатора по схеме RC цепочки с заданной точностью  ε = 10-3,  *Uвх*= 50 В:

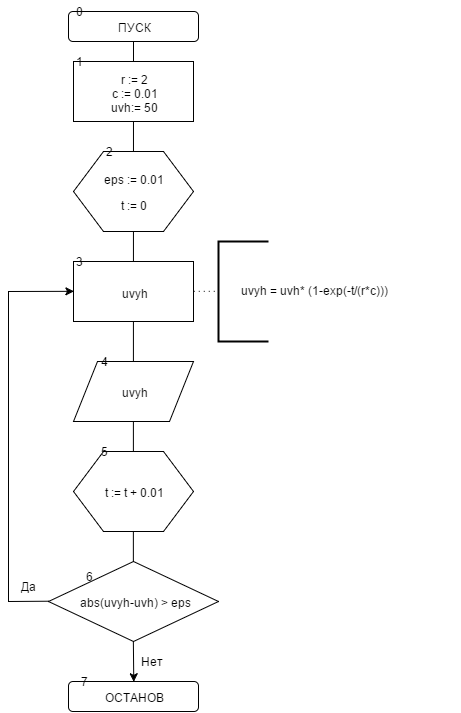


начальное значение *t*= 0, с шагом 0.01

**Математическая модель:**



**Блок-схема:**



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| r | real | Сопротивление |
| c | real | Ёмкость |
| uvh | real | Входное напряжение |
| uvyh | real | Выходное напряжение |
| t | real | Время |
| eps | real | Точность |

**Код программы:**

program zadanie1;

var

r,c, uvh, uvyh,t,eps : real;

begin

eps := 0.01;

r := 2;

c := 0.01;

t := 0;

uvh:= 50;

repeat

uvyh:= uvh \* (1-exp(-t/(r\*c)));

writeln('При t = ',t:2:2,' Выходное напряжение = ',Uvyh:2:6);

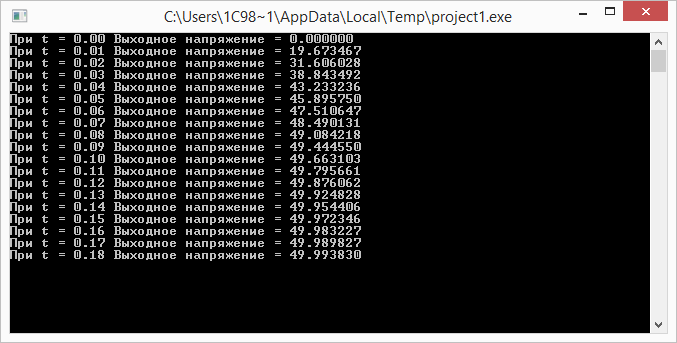
t := t + 0.01;

until abs(uvyh - uvh) <= eps;

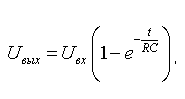
readln;

end.

**Результат работы программы:**



**Анализ:** Мы написали программу, которая строит переходную характеристику заряда конденсатора. По математической модели:



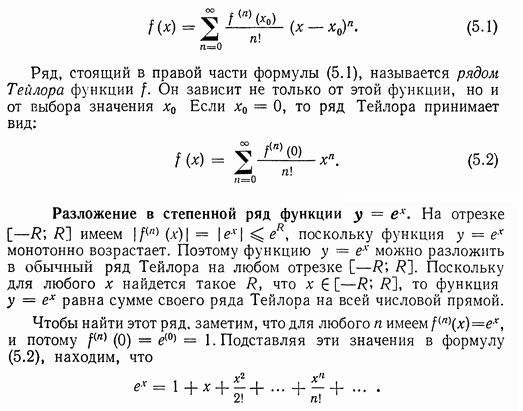
**Вывод:** Программа работает согласно математической модели.

**Задание 2.**

**Постановка задачи:**

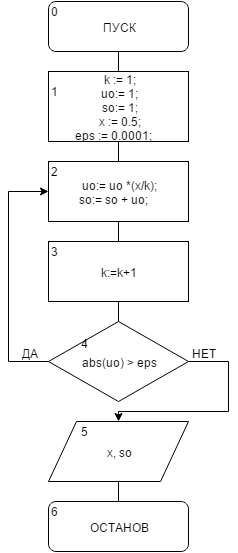
Вычислить e*x* с точность 10-4. Начальные условия: *k* = 1, *U0*= 1,  S0 = 1,  *x* = 0.5

**Математическая модель:**





**Блок-схема:**



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| uo | real | «Текущее» значение |
| so | real | «Следующее» значение |
| x | real | Степень |
| eps | real | Точность |
| fact | real | Факториал |
| k | integer | См. формулу |
| j | integer | Параметр цикла |

**Код программы:**

program task2;

var

uo, so,x,eps: real;

k : integer;

begin

k := 1;

uo:= 1;

so:= 1;

x := 0.5;

eps := 0.001;

while abs(uo) > eps do begin

uo:= uo \*(x/k);

so:= so + uo;

k := k+1;

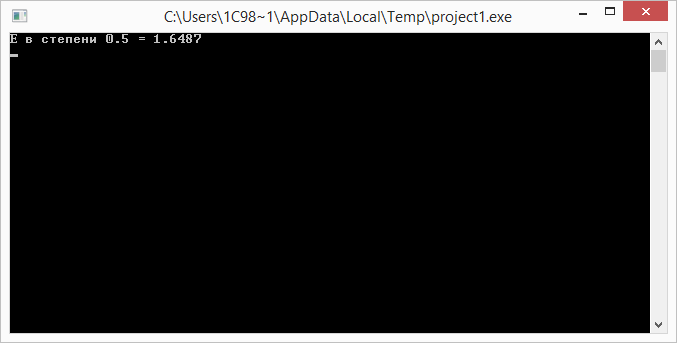
end;

writeln('Е в степени ',x:2:1,' = ', so:2:4);

readln;

end.

**Результат работы программы:**



**Анализ:** Мы написали программу, которая вычисляет по формуле Uo до тех пор, пока не достигнет заданной точности

**Вывод:** Мы написали программу, которая вычисляет по формуле



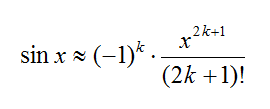
Uo до тех пор, пока не достигнет заданной точности

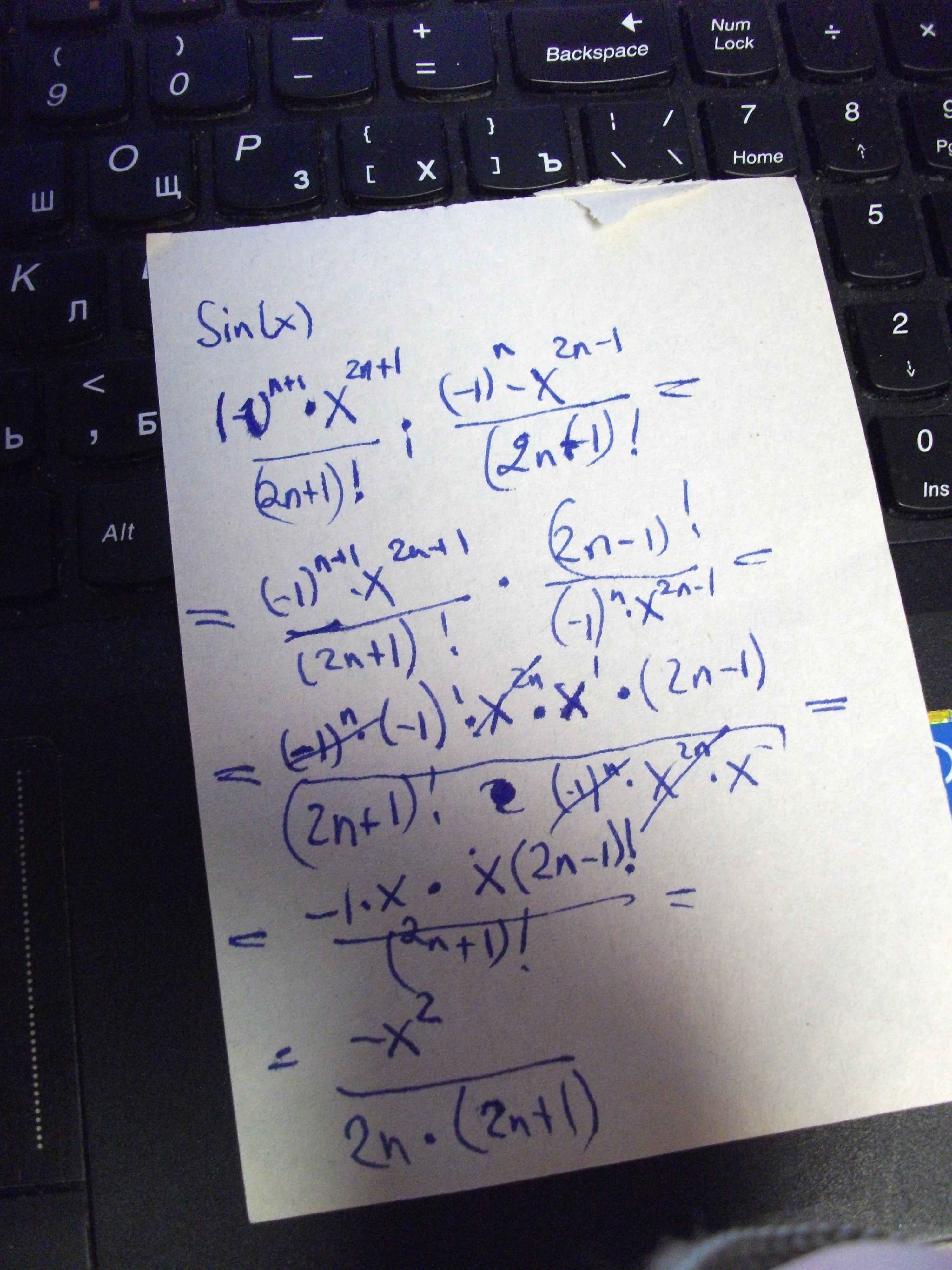
**Задание 3.**

**Постановка задачи:**

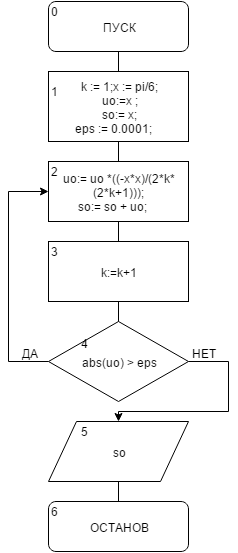
Вычислить Sin(*x*) с точностью 10-4. Начальные условия: k = 1,  U0 = x,  S0 = *x*,  x = π/6

**Математическая модель:**

****



**Блок-схема:**

****

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| uo | real | Текущее значение |
| so | real | Следующее значение |
| x | real | Выражение под знаком sin |
| eps | real | Точность |
| k | integer | См. формулу |

**Код программы:**

program zad3;

var

uo, so, x, eps: real;

k : integer;

begin

k := 1;

x := pi/6;

uo:= x;

so:= x;

eps := 0.0001;

while abs(uo) > eps do begin

uo:= uo \*((-x\*x)/(2\*k\*(2\*k+1)));

so:= so + uo;

k := k + 1;

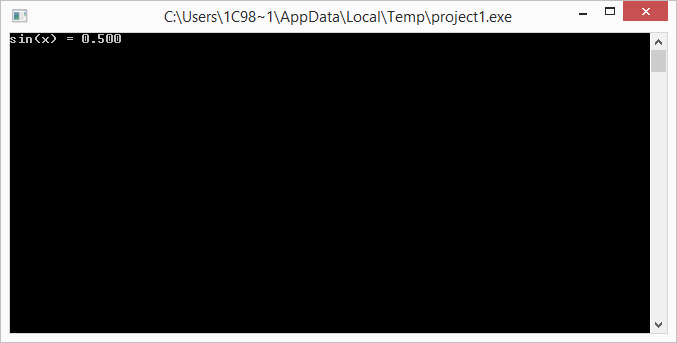
end;

writeln('sin(x) = ', so:2:3);

readln;

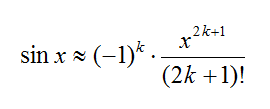
end.

**Результат работы программы:**



**Анализ:** Мы написали программу, которая вычисляет синус числа.

**Вывод:** Мы написали программу, которая вычисляет синус числа по мат.модели:

****

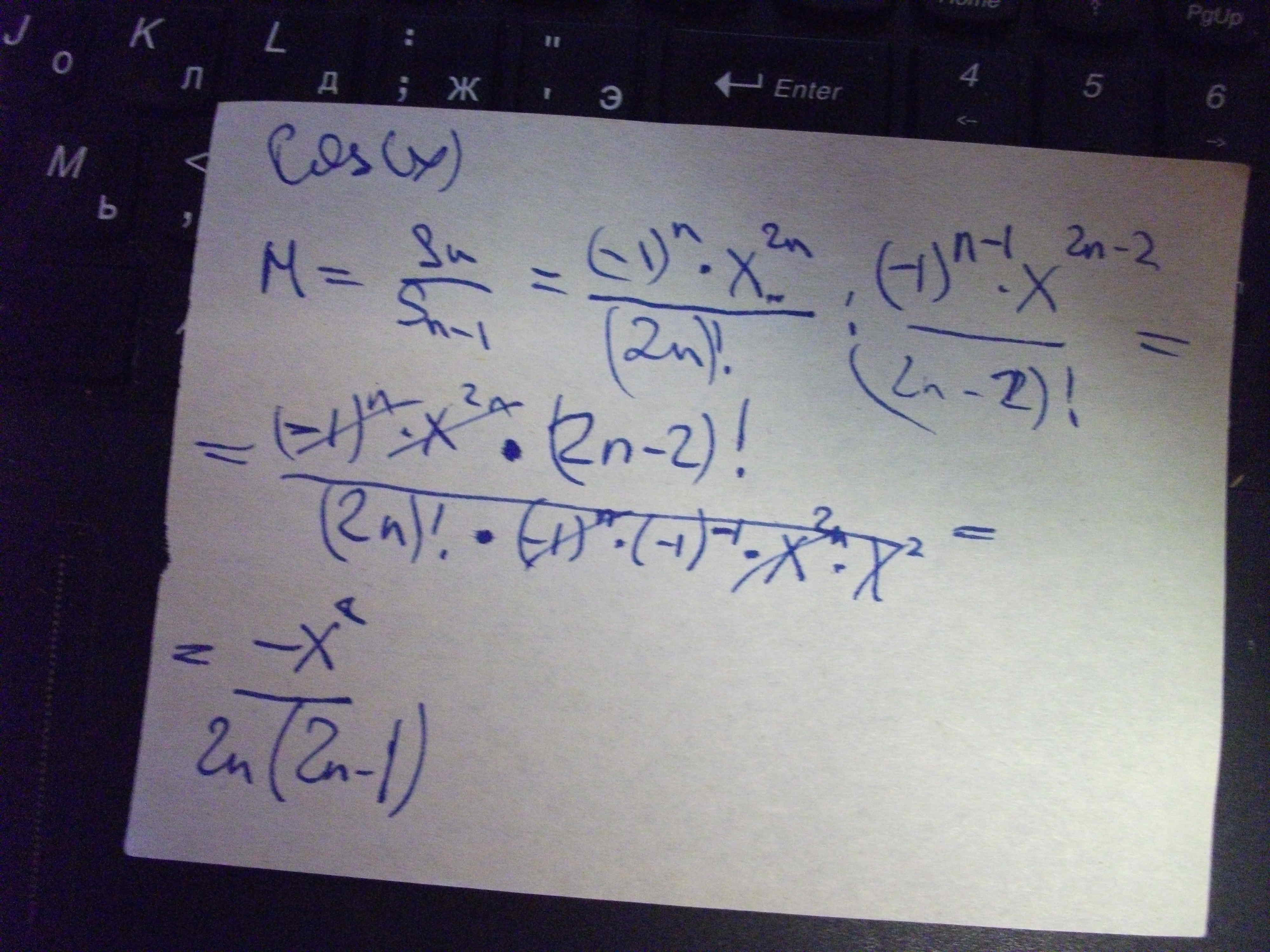
Результат верный.

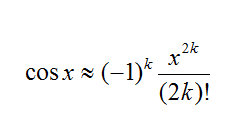
**Задание 4.**

**Постановка задачи:**

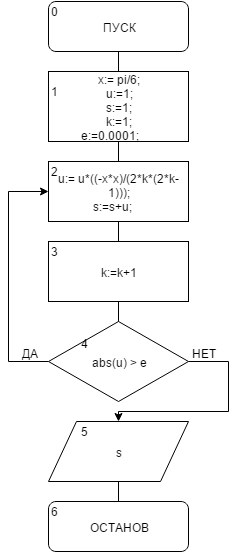
Вычислить Cos(*x*) с точностью10-4. Начальные условия: k = 1,  U0 = 1,  S0 = 1,  x = π/6

**Математическая модель:**



****

**Блок-схема:**

****

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| u | real | «Текущее» значение |
| s | real | «Следующее» значение |
| x | real | Выражение под знаком cos |
| e | real | Точность |
| k | integer | См. формулу |

**Код программы:**

program zad4;

var x, u, s, e :real;

k :integer;

begin

x:= pi/6;

u:=1;

s:=1;

k:=1;

e:=0.0001;

while abs(u) > e do

begin

u:= u\*((-x\*x)/(2\*k\*(2\*k-1)));

s:=s+u;

k:=k+1;

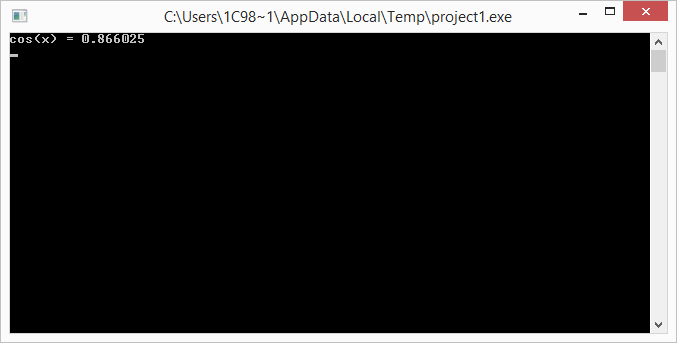
end;

writeln('cos(x) = ', s:0:6);

readln();

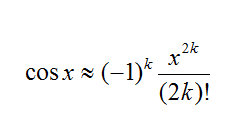
end.

**Результат работы программы:**



**Анализ:** Мы написали программу, которая считает косинус числа х

**Вывод:** Мы написали программу, которая считает косинус числа х по мат.модели:

****

Результат верный.