

Лабораторная работа №8.
ИЦВП с управлением по аргументу и функции.
Вычисление элементарных функций.

Цель: средствами языка С научиться работать с ИЦВП по аргументу и функции, вывести формулу для М и вычислить элементарные функции.

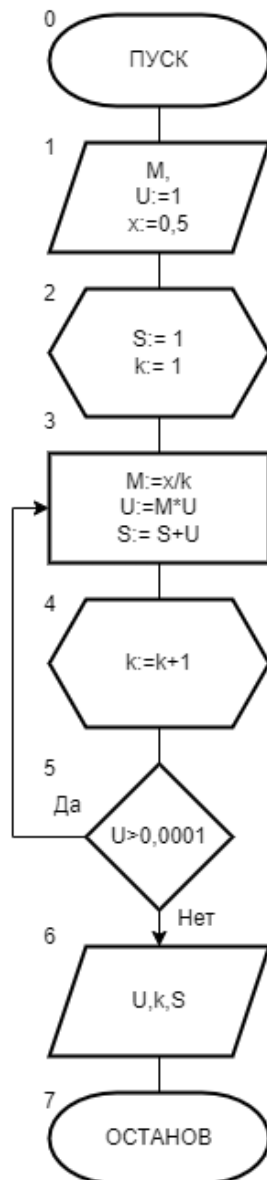
Используемое оборудование: ПК, Visual Studio Code

Задача 1: вычислить функцию $e^x \approx \sum_{k=0}^{\infty} \frac{x^k}{k!}$

Математическая модель:

$$U_k = M * U_{k-1}$$
$$M = \frac{x}{k}$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
U	Double	Данное значение
X	Double	Данное значение
C	Double	Данное значение
K	Double	Данное значение
M	Double	Рекуррентная зависимость
S	Double	Сумма (значение функции)

Код программы:

```
#include <stdio.h>
#include <math.h>

int main(void){

double M;
double U = 1;
double S = 1;
double x = 0.5;
double k = 1;

do
{
    M = x/k;
    U = M*U;
    S += U;
    k++;
} while (U > 0.0001);

printf("U: %lf\t", U);
printf("k: %lf\t", k);
printf("S: %lf", S);
}
```

Результат работы программы:

```
U: 0.000022    k: 7.000000    S: 1.648720
```

Анализ вычислений:

Были введены переменные, в цикле считаем M, после чего вычисляем U и каждое новое значение добавляем к переменной S.

Вывод:

средствами языка C я научился работать с ИЦВП по аргументу и функции и реализовал вычисление элементарной функции.

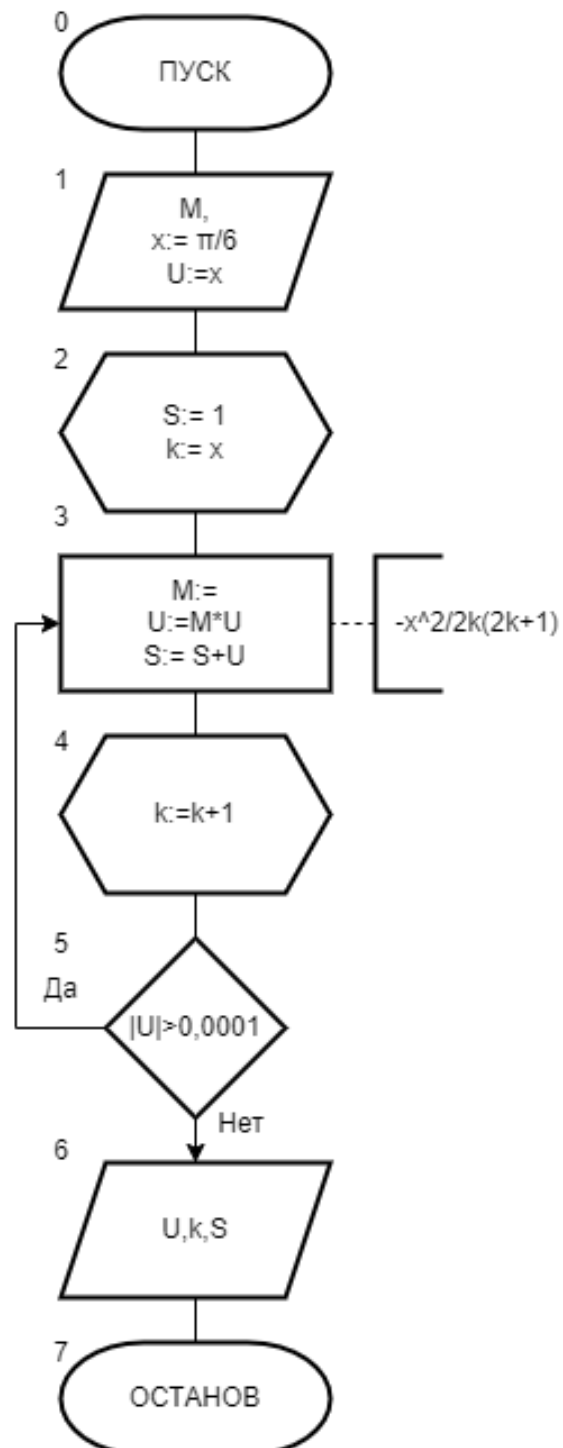
Задача 2: вычислить функцию $\sin x \approx \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$

Математическая модель:

$$U_k = M * U_{k-1}$$

$$M = \frac{-x^2}{2k(2k+1)}$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
U	Double	Данное значение
X	Double	Данное значение
C	Double	Данное значение
K	Double	Данное значение
M	Double	Рекуррентная зависимость
S	Double	Сумма (значение функции)

Код программы:

```
#include <stdio.h>
#include <math.h>

int main(void){

    double M;
    double x = 3.14159265/6;
    double U = x;
    double S = x;
    double k = 1;

    do
    {
        M = -(x*x)/(2*k*(2*k+1));
        U = M*U;
        S += U;
        k++;
    } while (fabs(U) > 0.0001);

    printf("U: %lf\t", U);
    printf("k: %lf\t", k);
    printf("S: %lf", S);
}
```

Результат работы программы:

```
U: -0.000002    k: 4.000000    S: 0.500000
```

Анализ вычислений:

Были введены переменные, в цикле считаем M, после чего вычисляем U и каждое новое значение добавляем к переменной S.

Вывод:

средствами языка C я научился работать с ИЦВП по аргументу и функции и реализовал вычисление элементарной функции.

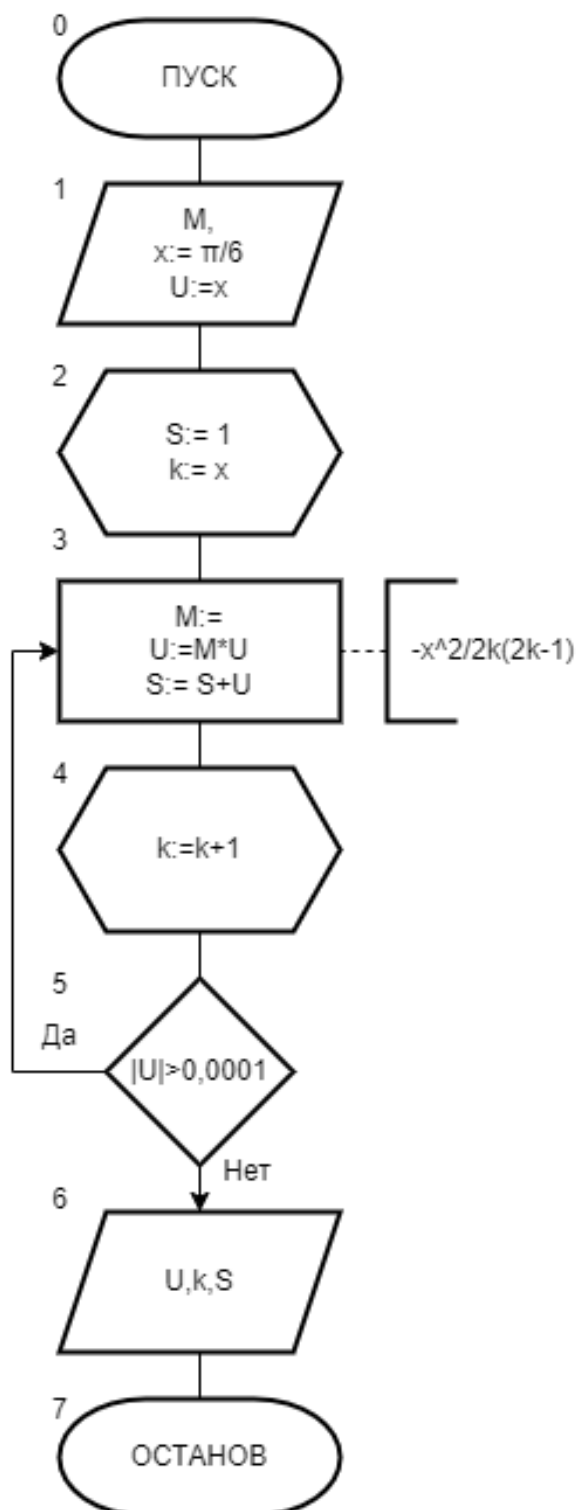
Задача 3: вычислить функцию $\cos x \approx \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!}$

Математическая модель:

$$U_k = M * U_{k-1}$$

$$M = \frac{-x^2}{2k(2k-1)}$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
U	Double	Данное значение
X	Double	Данное значение
C	Double	Данное значение
K	Double	Данное значение
M	Double	Рекуррентная зависимость
S	Double	Сумма (значение функции)

Код программы:

```
✓ #include <stdio.h>
#include <math.h>

✓ int main(void){

    double M;
    double x = 3.14159265/6;
    double U = 1;
    double S = 1;
    double k = 1;

    do
    {
        M = -(x*x)/(2*k*(2*k-1));
        U = M*U;
        S += U;
        k++;
    } while (fabs(U) > 0.0001);

    printf("U: %lf\t", U);
    printf("k: %lf\t", k);
    printf("S: %lf", S);
}
```

Результат работы программы:

```
U: -0.000029    k: 4.000000    S: 0.866025
```

Анализ вычислений:

Были введены переменные, в цикле считаем M, после чего вычисляем U и каждое новое значение добавляем к переменной S.

Вывод:

средствами языка C я научился работать с ИЦВП по аргументу и функции и реализовал вычисление элементарной функции.