

## Лабораторная работа 6.

### ИЦВП по функции.

Цель работы: средствами языка C научиться работать с ИЦВП по функции.

Используемое оборудование: ПК, Visual Studio Community.

Задача 1: С клавиатуры вводится трехзначное число, считается сумма его цифр.

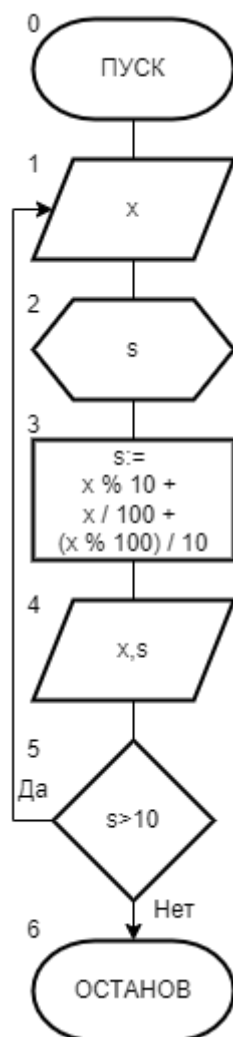
Если сумма цифр числа больше 10, то вводится следующее трехзначное число,

если сумма меньше либо равна 10 – программа завершается.

Математическая модель:

$$s = x \% 10 + \frac{x}{100} + \frac{x \% 100}{10}$$

Блок схема:



Список идентификаторов:

| Имя | Тип | Смысл              |
|-----|-----|--------------------|
| x   | int | Вводимое число     |
| s   | int | Сумма цифр числа x |

Код программы:

```
#include <iostream>

int main()
{
    int x,s;
    do {
        printf("Enter a three digit number:\n");
        scanf_s("%d", &x);
        s = x % 10 + x / 100 + (x % 100) / 10;
        printf("%d", s);
        printf("\n");
    } while (s > 10);
}
```

Анализ результатов вычислений:

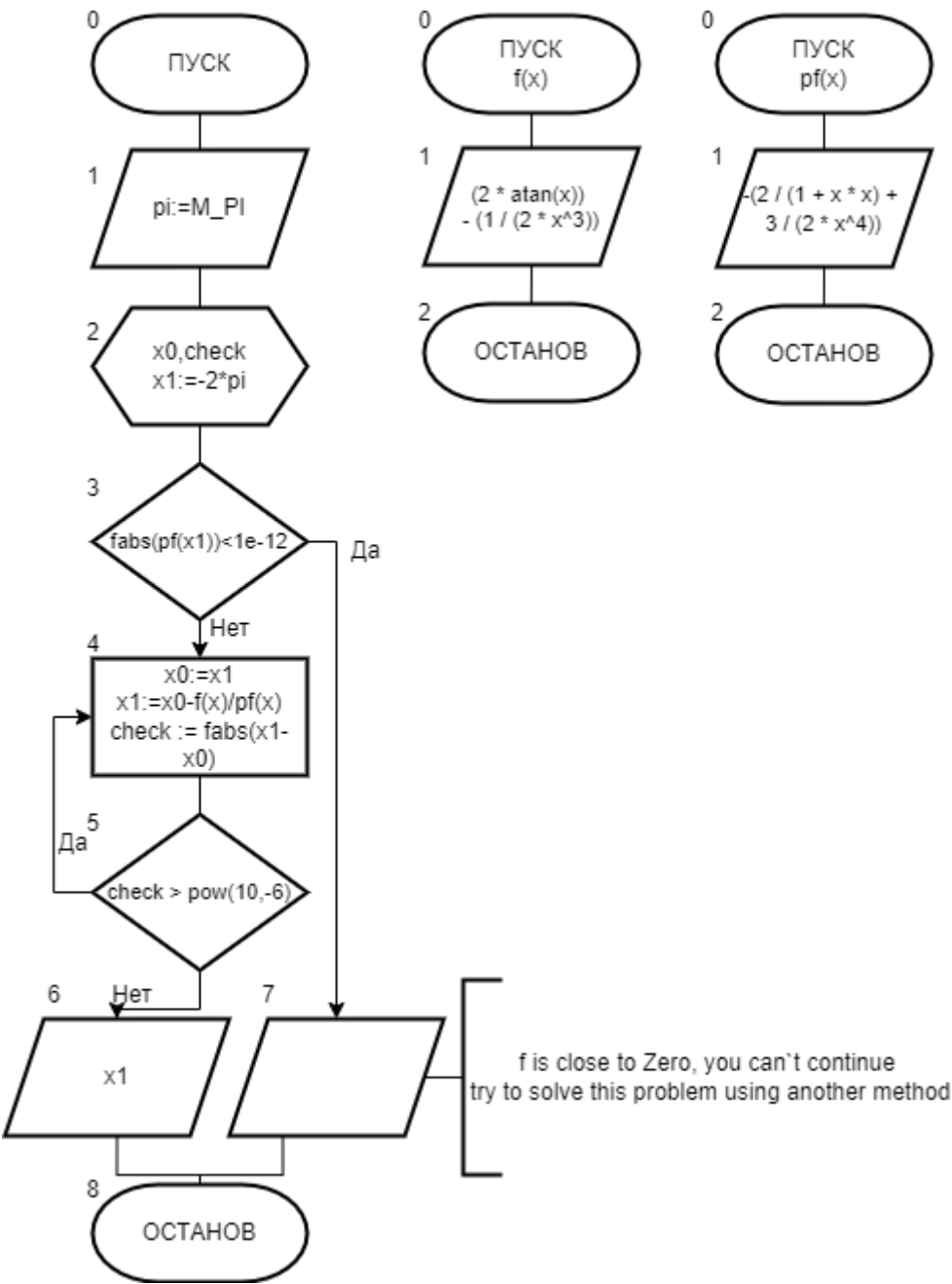
Были введены две переменные. В цикле с постусловием вводилось трехзначное число и считалась сумма цифр этого числа.

Вывод: средствами языка С научился работать с ИЦВП по функции.

Задача 2: Решить нелинейное уравнение  $2arctg(x) - \frac{1}{2x^3} = 0$  на отрезке от  $-2\pi$  до  $2\pi$  с точностью  $10^{-6}$  методом Ньютона.

Математическая модель:  $2arctg(x) - \frac{1}{2x^3} = 0$

Блок схема:



Список идентификаторов:

| Имя | Тип    | Смысл            |
|-----|--------|------------------|
| pi  | double | Число пи         |
| x0  | double | Начало интервала |
| x1  | double | Конец интервала  |

|       |        |   |
|-------|--------|---|
| check | double | Упрощение выражения для проверки точности |
| f     | double | Функция                                   |
| pf    | double | Производная функции                       |

Код программы:

```
#define _USE_MATH_DEFINES
#include <math.h>
#include <iostream>
double f(double x)
{
    return (2 * atan(x)) - (1 / (2 * pow(x,3)));
}

double pf(double x)
{
    return -(2 / (1 + x * x) + 3 / (2 * pow(x, 4)));
}

int main()
{
    double pi = M_PI;
    double x0, x1, check;
    x1 = -2 * pi;
    do {
        if (fabs(pf(x1)) < 1e-12) {
            printf("f is close to Zero, you can't continue\n try to solve this problem using another method");
            return 1;
        }
        x0 = x1;
        x1 = x0 - (f(x0) / pf(x0));
        check = fabs(x1 - x0);
    } while (check > pow(10,-6));
    printf("Answer: %lf\n", x1);
    return 0;
}
```

Результат работы программы:

```
f is close to Zero, you can't continue
try to solve this problem using another method
```

Анализ вычислений:

Попробовав различные варианты кода, я пришел к выводу, что к данному выражению метод Ньютона не применим, так как производная слишком мала или часто обращается в ноль.

Вывод: средствами языка С научился работать с ИЦВП по функции.