

Практическая работа №2

Тема: Алгоритмическая структура ветвление

Цель работы: Построить графический алгоритм и решить задачи по тема ветвление и цикл

Теоретические сведения:

Алгоритмическая структура ветвление

В языке Си, как и в любом другом структурированном языке программирования, помимо конструкции последовательного выполнения операторов, есть еще две фундаментальные конструкции: **ветвление и цикл**.

Стоит отметить, что при определенном сочетании значений, получим отрицательное подкоренное выражение, поэтому для решения подобных задач целесообразно применение оператора ветвление

Ветвление используется тогда, когда возникает необходимость изменить ход выполнения программы добавлением развилки, разбивающей его на две и более альтернативных ветви.

Синтаксис конструкции ветвления следующий:

if (условие) <оператор> Условием является любое выражение. Как правило, применяется логическое выражение, *но язык Си позволяет использовать и любое арифметическое, трактуя его нулевое значение как ложь, а любое отличное от нуля – как истину.*

Логика выполнения конструкции очень простая – если условие истинно, то выполняется идущий следом <оператор>, в противном случае он пропускается. В качестве <оператор> может быть любой оператор: одиночный (если значение a отрицательно, то мы его делаем положительным): `if (a < 0) a = a * -1;` составной (если a больше b , то меняем значения переменных местами): `if (a > b) { int tmp = a; a = b; b = tmp; }` и даже пустой.

Конструкция `if (условие) <оператор_1>; else <оператор_2>` формируется без знака «;» перед `else`.

Логика работы этой конструкции следующая: если условие истинно, то выполняется <оператор_1>, а <оператор_2> пропускается, а если условие ложно, то все наоборот: <оператор_1> пропускается, а <оператор_2> выполняется.

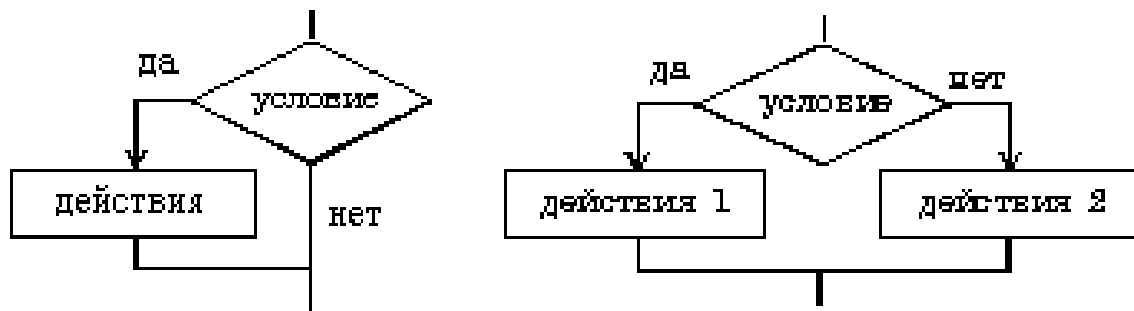


Рис. 2 Ветвление

Примеры ветвления

Задание: Написать программу, спрашивающую у пользователя два числа и выводящую «TRUE», если первое меньше второго. В противном случае вывести «FALSE».

Решение :

```

#include <iostream>
#include <locale>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    int a, b;
    cout << "Введите первое число a" << endl;
    cin >> a;
    cout << "Введите второе число b" << endl;
    cin >> b;
    if (a < b) cout << "True"; else cout << "False";
    return 0;
}

```

Задание Найти максимум среди трех чисел.

Решение:

```

#include <iostream>
#include <locale>
using namespace std;
int main()
{
    setlocale(0, "");
    int a, max, b, c;

    cout << "Введите 3 числа" << endl;
    cin >> a;
    cin >> b;
    cin >> c;

    if (a >= b && a >= c)
        max = a;
    else {
        if (b >= a && b >= c) max = b;
    else max = c;
    }
    cout << "Максимум среди трех чисел :" << max;
    return 0;
}

```

Алгоритмическая структура цикл

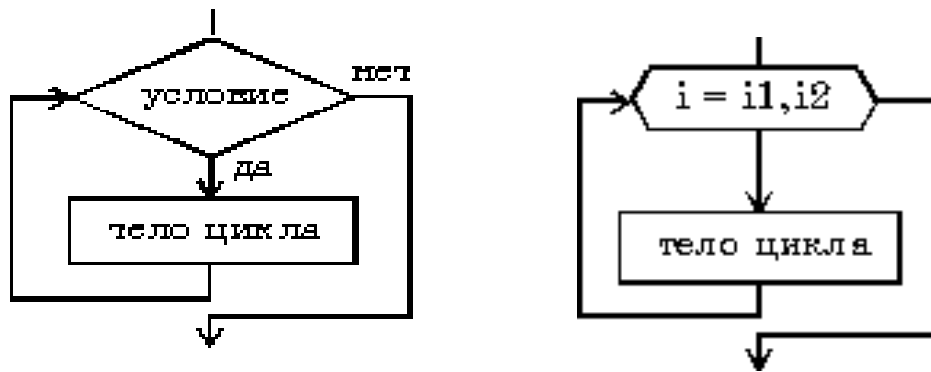


Рис. 3 Цикл

Если возникает необходимость повторить какой-то оператор некоторое количество раз, используется третья конструкция – цикл.

Существуют две разновидности циклов – управляемые условием и управляемые счетчиком. В свою очередь первые делятся на циклы с предусловием и постусловием. Ключевыми словами для таких конструкция являются:

`while <условие> <тело цикла>;` в данном случае сначала осуществляется проверка условия и если условие верно (`true` или больше 1) выполняются операторы тела цикла.

Вторая конструкция получила название цикл с постусловием:

`do <тело цикла>`

`while <условие>;`

Пример использования цикла с предусловием:

Вычислите факториал числа `n`:

Решение:

```
#include <iostream>;
#include <locale>;
using namespace std;

int main()
{
    setlocale(0, "");
    int i, n, fact;
    cout << "Введите число n" << endl;
    cin >> n;
    i = 1; fact = 1;
    while (i < n) {
        fact *= i;
        i++;
    }
    cout << n << "!=" << fact;
    return 0;
}
```

Часто цикл с постусловием применяют для проверки правильности ввода:
Напишите программу, которая запрашивает у пользователя целое число в диапазоне от 0 до 10. Программа должна запрашивать число до тех пор, пока пользователь не введет его из нужного диапазона. После этого программа печатает на экране «Ok» и заканчивает работу.

Решение :

```
#include <iostream>
#include<locale>
using namespace std;

int main()
{
    setlocale(0, "");
    int i;
    do {
        cout << "Введите целое число от 0 до 10" << endl;
        cin >> i;
    }

        while (i < 0 || i>10);

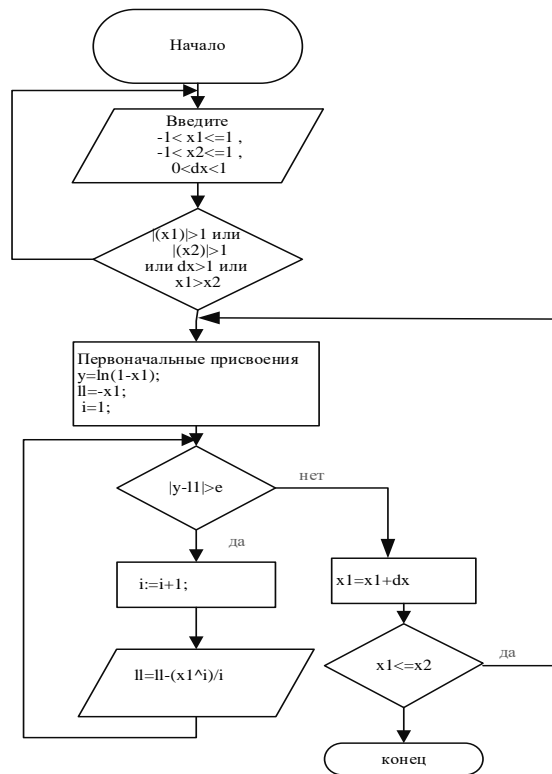
    cout << "Спасибо!";
    return 0;}
```

Рассмотрим еще пример применения циклов для расчета выражений с помощью ряда Тейлора.

Задание вычислить функцию, заданную с помощью ряда Тейлора, на интервале от x_1 до x_2 с шагом dx и с точностью $\epsilon=0,0001$. Ввод x осуществить с клавиатуры, учесть вероятность неправильного ввода.

$$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} = -\left(x + \frac{x^2}{2} + \frac{x^4}{4} + \dots\right), \quad -1 \leq x < 1.$$

Построим графический алгоритм решения данной задачи. Для этих целей можно, например, использовать Visio или специализированные ПП.



```

#include <iostream>
#include <locale>
#include <cmath> // подключение библиотеки
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");
    float x1, x2, dx;
    float const e = 0.01;
    cout << "Программа вычисления линейного логарифма для диапазона x1..x2 с  
заданной точностью" << endl;
    do {
        cout << "Введите число -1<x1<=1" << endl;
        cin >> x1;
        cout << "Введите число -1<x2<=1 x2>x1" << endl;
        cin >> x2;
        cout << "Введите шаг вычислений 0<dx<1" << endl;
        cin >> dx;
    }
    while (abs(x1) >= 1 || abs(x2) >= 1 || dx >= 1 || x1 > x2);
    cout.precision(4);
    float y, l1;
    int i;
    do
    {
        y = log(1 - x1);
        l1 = -x1; i = 1;
        while (abs(y - l1) > e)
        {
            i++;
            l1 = l1 - (pow(x1, i)) / i;
        }
        cout << "ln (" << x1 << ")=" << l1;
        x1 = x1 + dx;
    }
    while (x1 <= x2);
    return 0;
}

```

Циклы с счетчиком применяют, если количество повторов заранее известно. Такие циклы наиболее востребованы, например для ввода/ вывода массивов чисел, вычисления рядов и т. п.

Конструкция цикла for:

```
for (<инициализация>; <условие>; <инкремент>)  
<тело цикла> ;
```

Приведем простейший пример использования цикла for:

Пример : Напечатать все числа от 1 до 10, разделенные пробелом.

Решение :

```
int i;  
for (i = 1; i <= 10; i++) {  
    cout << i << " ";  
}
```

Операторы передачи управления в C++

При работе с циклами часто используются операторы передачи управления.

четыре оператора, изменяющих естественный порядок выполнения вычислений:

- оператор безусловного перехода goto;
- оператор выхода из цикла break;
- оператор перехода к следующей итерации цикла continue;
- оператор возврата из функции return.

Оператор goto

Оператор безусловного перехода goto имеет формат:

```
goto <метка>;.....
```

```
<метка>: <оператор>;
```

С точки зрения структурного программирования : следование ветвление и цикл, применение данного оператора считается нежелательным но не запрещается. Подобные переходы могут вызвать ошибки, которые в дальнейшем сложно отследить и выявить например:

```
goto metka;  
{  
int a = 3, b = 4;  
k= a + b;  
metka: int m = k + 1; ...
```

}

В приведенном фрагменте вроде нет синтаксических ошибок, но после его выполнения m -не определено.

Оператор break.

Оператор break используется внутри операторов цикла или switch для обеспечения перехода в точку программы, находящуюся непосредственно за оператором, внутри которого находится break.

Оператор continue

Оператор перехода к следующей итерации цикла continue пропускает все операторы, оставшиеся до конца тела цикла, и передает управление на начало следующей итерации.

Оператор return

Оператор возврата из функции return завершает выполнение функции и передает управление в точку ее вызова. Вид оператора: return [выражение]; Выражение должно иметь скалярный тип. Если тип возвращаемого функцией значения описан как void, выражение должно отсутствовать.

Варианты заданий по теме ветвление

Разветвляющие вычислительные процессы

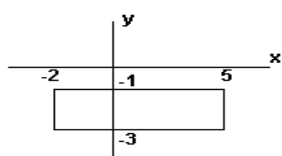
1. Вычислить значение:

$$a = z + 4z^2, \quad \text{если } z = \begin{cases} y^2 - 1, & \text{при } y < 0 \\ 0, & \text{при } 0 \leq y \leq 1 \\ y^2 + 3y, & \text{при } y > 1 \end{cases};$$

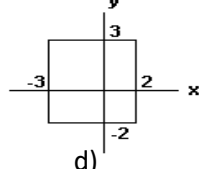
2. Вычислить значение: $m = s + 2y$, если $s = \begin{cases} y - 1, & \text{при } y < 0 \\ 0, & \text{при } y \geq 0 \end{cases}$;

3. Вычислить значение функции при заданном значении аргумента x : $y = \frac{1}{x} + \frac{1}{x-1}$;

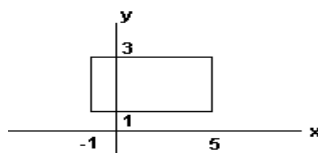
4. На плоскости xOy задана фигура. Определить, принадлежит ли заданная точка $M(x,y)$ данной фигуре. Задачу решить двумя способами, используя оператор ветвления и оператор case.



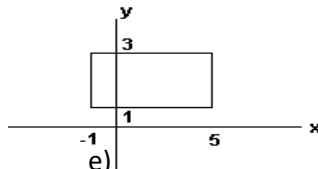
a)



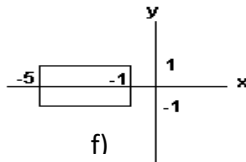
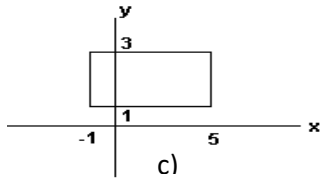
d)



b)



e)



5. Вычислить значение функции при заданном значении аргумента x :

$$y = \frac{1}{x-2} + 2x;$$

6. Вычислить значение функции при заданном значении аргумента x :

$$y = \frac{1}{x} + 2x^2;$$

8. Вычислить значение: $a = z + 4z^2$, если $z = \begin{cases} y^2 - 1, & \text{при } y < 0 \\ 0, & \text{при } 0 \leq y \leq 1 \\ y^2 + 3y, & \text{при } y > 1 \end{cases}$;

Варианты заданий по теме цикл

1. Вычислить сумму $Z=1+3+5+7+\dots$. Вычисления прекратить, когда Z превысит заданное значение A .

2. Вычислить сумму $Z=1+2+3+\dots$. Вычисления прекратить, когда Z превысит заданное значение A .

3. Вычислить сумму заданного количества членов: $S = \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$

4. Вычислить сумму: $S = 1 + \frac{1}{3} + \frac{1}{5} + \dots$. Вычисления прекратить, когда S превысит заданное значение A .

5. Вычислить сумму $S = \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$. Вычисления прекратить, когда S превысит заданное значение A .

6. Составьте программу вычисления суммы вида: а) $1 + \frac{1}{2^3} + \frac{1}{3^3} + \dots + \frac{1}{50^3}$; б)

$$\frac{1}{2^2} + \frac{1}{4^2} + \dots + \frac{1}{128^2}$$

7. Пусть дано 100 вещественных чисел, которые генерируются случайным образом. Найдите количество положительных, отрицательных и нулевых чисел.

8. Пусть дано 100 вещественных чисел, которые генерируются случайным образом. Вычислите среднее арифметическое положительных и среднее арифметическое отрицательных чисел.

9. Пусть дано 100 вещественных чисел, которые генерируются случайным образом. Вычислите разность между максимальным и минимальным числами.

10. Напишите программу нахождения первого члена последовательности который не принадлежит заданному отрезку $[a, b]$. $a_n = (-1)^n (1 + \frac{1}{2} + \dots + \frac{1}{n})$, где $n=1,2,\dots$

Варианты заданий по теме ряд Тейлора

Определение приближенного значения функции заданной с помощью ряда Тейлора

1. Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x_1 до x_2 с шагом dx с точностью $\epsilon=0,0001$. Значения x_1 , x_2 , шаг dx вводятся пользователем с клавиатуры.

$$1. \ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} = -(x + \frac{x^2}{2} + \frac{x^4}{4} + \dots), -1 \leq x \leq 1$$

$$2. \ln \frac{x+1}{x-1} = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = 2(\frac{1}{x} - \frac{1}{3x^3} + \frac{1}{5x^5} + \dots) \quad |x| < 1$$

$$3. \operatorname{arctg}(x) = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1} x^{2n+1}}{2n+1} = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} \dots \quad |x| \leq 1$$

$$4. \ln(x+1) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{n+1} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots - 1 < x \leq 1$$

$$5. \operatorname{arctg}(x) = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2m+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} \dots \quad x > 1$$

$$6. \operatorname{arctg}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} \dots \quad |x| \leq 1$$

$$7. e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots x < \infty$$

$$8. \quad \frac{\sin x}{x} = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n+1)!} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots x < \infty$$

$$9. \quad \cos x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots |x| < \infty$$