**Лабораторная работа №4. Разработка алгоритмов и программ циклической структуры**

**Циклический алгоритм** - это алгоритм, который предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же математическим зависимостям, но при разных значениях некоторой специально изменяемой величины. Многократное выполнение некоторой группы повторяющихся операторов называют *телом цикла.* Специально изменяемый по заданному закону параметр, входящий в тело цикла, *называют переменной цикла.*Переменная цикла используется для подготовки очередного повторения цикла и отслеживания условий его окончания. Во время выполнения тела цикла параметры переменной цикла изменяются в интервале от начального до конечного значения с заданным шагом.

При организации циклических вычислений необходимо предусмотреть следующее: начальное значение переменной цикла; конечное значение переменной цикла; закон изменения переменной цикла перед каждым новым его повторением, т.е. необходимо предусмотреть шаг цикла.

Цикл называется **простым,** если в его теле нет разветвлений и других встроенных циклов.

Цикл называется **сложным***,* если в его теле есть разветвления и (или) другие встроенные в цикл циклы.

Цикл называется **детерминированным,** если в нем число повторений заранее известно из исходных данных или определено в ходе решения задачи.

Цикл называется **итерационным***,* если в нем число повторений неизвестно из исходных данных и не определено по ходу решения задачи.

**Структура и типы операторов циклов**

Операторы цикла используются для многократно повторяющихся вычислений. Выделяют циклы с **известным** числом шагов и циклы с **неизвестным** числом шагов (циклы с условием).

**Цикл с параметром**

Синтаксис цикла с параметром:

**for ( инициализация; выражение; модификации ) оператор;**

{... тело цикла; …}

Инициализация служит для объявления величин, используемых в цикле, и присвоения им начальных значений. В этой части можно записать несколько операторов, разделенных запятой, например:

for ( int i = 0; j = 20; …

int k, m;

for ( k = 1; m = 0; …

Областью действия переменных, объявленных в части инициализации цикла, является цикл. Инициализация выполняется один раз в начале исполнения цикла. Выражение типа bool определяет условие выполнения цикла: если его результат равен true, цикл выполняется. Цикл с параметром реализован как цикл с предусловием. Модификации выполняются после каждой итерации цикла и служат обычно для изменения параметров цикла. В части модификаций можно записать несколько операторов через запятую, например:

for ( int i = 0; j = 20; i < 5 && j > 10; i++; j--) …

Простой или составной оператор представляет собой тело цикла. Пример использования цикла с параметром

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{ class Program

{

static void Main(string[] args)

{

int sum = 0;

int fact = 1;

Console.WriteLine("Вычислим факториал от 1 до 5");

for (int i = 1; i <= 5; i++)

{ sum += i;

fact \*= i; }

Console.WriteLine("Сумма равна: " + sum);

Console.WriteLine("Факториал равен: " + fact);

Console.ReadKey(); } } }

**Особенности:**

• ***условие*** проверяется в начале очередного шага цикла, если оно ложно, цикл не выполняется;

• ***изменения*** (третья часть в заголовке) выполняются в конце очередного шага цикла;

• если ***условие*** никогда не станет ложным, цикл может продолжаться бесконечно (**зацикливание**): **for(i=1; i<8; i++) { i--; }**

• **Не рекомендуется менять переменную цикла в теле цикла!**

• если в теле цикла один оператор, **скобки** {} можно не ставить: **for (i = 1; i < 8; i++) a += b.**



Рисунок 4.1 – Пример работы цикла с параметром

**Цикл с пред условием:**

Синтаксис оператора while:

**while ( выражение ) оператор**

Выражение должно быть логического типа. Если результат вычисления выражения равен true, выполняется простой или составной оператор (блок). Эти действия повторяются до того момента, пока результатом выражения не станет значение false. После окончания цикла управление передается на следующий за ним оператор.

Рекуррентная формула – формула, в которой новое значение переменной вычисляется с использованием ее предыдущего значения. Если в теле цикла необходимо выполнить более одного оператора, необходимо заключить их в блок с помощью фигурных скобок.

Пример использования цикла while

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{ class Program

{ static void Main(string[] args)

{ Console.WriteLine("Данный цикл будет выполняться пока х не будет равен 0");

Console.WriteLine("х=30 с");

int x=30;

while (x > 0)

{ x--;

if (x == 0)

{Console.Write("Ваше время истекло! х = " + x); } }

Console.ReadKey();

} } }

**Особенности:**

* можно использовать сложные условия: **while ( a < b && b < c ){ ... }**
* если в теле цикла только один оператор, скобки **{}** можно не писать: **while ( a < b ) a ++;**
* условие пересчитывается **каждый раз** при входе в цикл
* если условие на входе в цикл ложно, цикл не выполняется ни разу **a = 4; b = 6; while ( a > b ) a = a– b;**
* если условие никогда не станет ложным, программа **зацикливается a = 4; b = 6; while ( a < b ) d = a + b;**



Рисунок 4.2 – Пример работы цикла с пред условием

На рисунке 4.3 приведены особенности замены цикла с предусловием на цикл с параметром и наоборот.



Рисунок 4.3 – Особенности замены цикла с предусловием на цикл с параметром и наоборот

**Цикл с постусловием**

Синтаксис оператора do while:

**do оператор while выражение;**

Сначала выполняется простой или составной оператор, образующий тело цикла, а затем вычисляется выражение (оно должно иметь тип bool). Если выражение истинно, тело цикла выполняется еще раз и проверка повторяется. Цикл завершается, когда выражение станет равным false или в теле цикла будет выполнен какой - либо оператор передачи управления. Этот вид цикла применяется в тех случаях, когда тело цикла необходимо обязательно выполнить хотя бы один раз. Если такой необходимости нет, лучше пользоваться циклом с предусловием.

Пример использования цикла do while

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{ class Program

{ static void Main(string[] args)

{ int x = 0;

string y;

Console.WriteLine("Данная программа будет шагать вперед на 1 см, пока не достигнет нужной нам цели");

Console.Write("Введите кол-во шагов:");

y = Console.ReadLine();

int Y = Convert.ToInt32(y);

do

{ x++;

}while (x != Y);

if (x == Y)

{Console.WriteLine("Ну вот мы и пришли в столовую! =)Всем приятного аппетита!"); }

Console.ReadKey(); } } }

**Особенности:**

* тело цикла всегда выполняется хотя бы один раз;
* после слова **while** («пока…» ) ставится условие продолжения цикла.



Рисунок 4.4 – Пример работы оператора цикла с пост условием

**Задания для выполнения работы**

Выполните каждое задание, используя каждый из трех видов циклических структур.

Вариант 1.

1. Дано натуральное число n. Найти среднее арифметическое n чисел, вводимых с клавиатуры.

2. Даны n положительных и отрицательных чисел и действительное число m. Найти среднее арифметическое чисел, меньших *т,* и среднее геометрическое чисел, больших либо равных *т.*

Вариант 2.

1. Дано натуральное число n. Найти среднее геометрическое n чисел, вводимых с клавиатуры.

2. Написать программу, возводящую число 2 в степени от 2 до 10 включительно.

Вариант 3.

1. Определить количество трехзначных натуральных чисел, сумма цифр которых равна заданному числу n.

2. Дано натуральное четырехзначное число *п.* Выяснить, входит ли цифра 3 в запись этого числа.

Вариант 4.

1. Среди двухзначных чисел найти те, сумма квадратов цифр которых делится на 13.

2. Дано вещественное число *а* и целое число *п (п >* 0) Написать программу, возводящую число *а* в степень от 1 до n включительно.

Вариант 5.

1. Написать программу поиска двухзначных чисел таких, что если к сумме цифр этого числа прибавить квадрат этой суммы, то получится это число.

2. Дано вещественное число *а (а* >1). Вывести наименьшее из целых чисел n, для которых сумма: 1+ 1/2+1/3+...+1/n будет больше *а.*

Вариант 6.

1. Написать программу поиска четырехзначного числа, которое при делении на 133 дает в остатке 125, а при делении на 134 дает в остатке 111.

2. Даны 10 действительных чисел. Вычислить сумму чисел, больших 100 и вывести эти числа.

Вариант 7.

1. Найти сумму целых положительных чисел из промежутка от a до b, кратных 4.

2. Ввести два натуральных числа *х* и *у (х <у).* Увеличивать первое число на 1, пока оно не будет равно второму. Вывести оба числа на экран.

Вариант 8.

1. Найти сумму целых положительных чисел, больших 20, меньших 100, кратных 3 и заканчивающихся на 2, 4 или 8.

2. Ввести три числа. Увеличивать третье число в 10 раз, пока оно не будет больше произведения двух других. Вывести на экран три числа.

Вариант 9.

1. Дано натуральное число n. Получить все его натуральные делители.

2. Ввести число. Увеличивать его на 1, пока обратное от этого числа не будет меньше 0,001 (обратное число - 1/х). Вывести на экран число и обратное от него.

Вариант 10.

1. Даны натуральные числа m и n. Получить все кратные им числа, меньшие m\*n.

2. Дано число a. Написать программу получения в порядке убывания всех делителей данного числа.

Вариант 11.

1. Сумма цифр трехзначного числа кратна 7, само число также делится на 7. Найти все такие числа.

2. Вводится целое число. Определить число четных цифр в данном числе.

Вариант 12.

1. Дано натуральное число n. Определить, является ли оно палиндромом (перевертышем) с учетом четырех цифр. Например, палиндромами являются следующие числа: 1111, 6116, 0440.

2. Вывести число. Увеличивать его в два раза, пока квадратный корень из числа не будет больше 10. Вывести на экран квадратный корень.

Вариант 13.

1. Даны натуральные числа n и k. Из чисел от n до k выбрать те, запись которых содержит ровно три одинаковые цифры. Например, числа 6766, 5444 содержат ровно три одинаковые цифры.

2. Дано вещественное число *а* и целое число *п (п >* 0) Написать программу, возводящую число *а* в степень от 1 до n включительно.

Вариант 14.

1. Среди четырехзначных чисел выбрать те, у которых все четыре цифры различны.

2. Ввести три числа. Увеличивать третье число в 10 раз, пока оно не будет больше произведения двух других. Вывести на экран три числа.

Вариант 15.

1. Дано четырехзначное число n. Выбросить из его записи цифры 0 и 5, оставив прежним порядок остальных цифр. Например, из числа 1509 должно получиться число 19.

2. Ввести число. Увеличивать его на 1, пока обратное от этого числа не будет меньше 0,001 (обратное число - 1/х). Вывести на экран число и обратное от него.

Вариант 16.

1. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенная в n-ю степень, равна самому числу (например, 153=13+53+33). Получить все трехзначные числа Армстронга.

2. Найти сумму всех чисел из промежутка от a до b, кратных 13 и 5.

Вариант 17.

1. Поменять местами первую и последнюю цифры числа.

2. Дано число a. Написать программу получения в порядке убывания всех делителей данного числа.

Вариант 18.

1. Поменять порядок цифр числа на обратный. Например, было 12345, стало 54321.

2. Если мы сложим все цифры какого-либо числа, затем все цифры найденной суммы и будем повторять много раз, мы наконец получим однозначное число, называемое цифровым корнем данного числа. Например, цифровой корень числа 34697 равен 2 (3+4+6+9+7=29; 2+9=11; 1+1=2). Составить программу для нахождения цифрового корня натурального числа n.

Вариант 19.

1. Найти количество четных цифр целого положительного числа.

2. Вводится целое число. Определить число четных цифр в данном числе.

Вариант 20.

1. Найти самую большую цифру целого числа.

2. Ввести два натуральных числа *х* и *у (х <у).* Увеличивать первое число на 1, пока оно не будет равно второму. Вывести оба числа на экран.

Вариант 21.

1. Найти сумму цифр целого числа, больших 5.

2. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенная в n-ю степень, равна самому числу (например, 153=13+53+33). Получить все трехзначные числа Армстронга.

Вариант 22.

1. Сколько раз данная цифра k встречается в числе n.

2. Ввести число. Увеличивать его на 1, пока обратное от этого числа не будет меньше 0,001 (обратное число - 1/х). Вывести на экран число и обратное от него.

Вариант 23.

1. Составить программу, проверяющую, является ли заданное натуральное число совершенным, т.е. равным сумме своих положительных делителей, кроме самого этого числа.

2. Найти все трехзначные числа при условии, что сумма его цифр равна a, а само число оканчивается цифрой b.

Вариант 24.

1. Найти натуральное число от 1 до n с максимальной суммой делителей.

2. Если мы сложим все цифры какого-либо числа, затем все цифры найденной суммы и будем повторять много раз, мы наконец получим однозначное число, называемое цифровым корнем данного числа. Например, цифровой корень числа 34697 равен 2 (3+4+6+9+7=29; 2+9=11; 1+1=2). Составить программу для нахождения цифрового корня натурального числа n.

Вариант 25.

1. Даны натуральные числа a и b (a<b). Получить все простые числа из промежутка от a до b.

2. Вводится целое число. Определить число нечетных цифр в данном числе.

Вариант 26.

1. Даны натуральные числа n и m. Найти все пары дружественных чисел, лежащих в диапазоне от n до m. Два числа называются дружественными, если каждое из них равно сумме всех делителей другого (само число в качестве делителя не рассматривается).

2. Дано вещественное число *а (а* >1). Вывести наименьшее из целых чисел n, для которых сумма: 1+ 1/2+1/3+...+1/n будет больше *а.*

Вариант 27.

1. Если мы сложим все цифры какого-либо числа, затем все цифры найденной суммы и будем повторять много раз, мы наконец получим однозначное число, называемое цифровым корнем данного числа. Например, цифровой корень числа 34697 равен 2 (3+4+6+9+7=29; 2+9=11; 1+1=2). Составить программу для нахождения цифрового корня натурального числа n.

2. Ввести два натуральных числа *х* и *у (х <у).* Увеличивать первое число на 1, пока оно не будет равно второму. Вывести оба числа на экран.

Вариант 28.

1. Найти все трехзначные числа при условии, что сумма его цифр равна a, а само число делится на b.

2. Дано число a. Написать программу получения в порядке убывания всех делителей данного числа.

Вариант 29.

1. Найти все трехзначные числа при условии, что сумма его цифр равна a, а само число оканчивается цифрой b.

2. Вывести число. Увеличивать его в два раза, пока квадратный корень из числа не будет больше 10. Вывести на экран квадратный корень.

Вариант 30.

1. Найти все четырехзначные числа, у которых сумма крайних цифр равна сумме средних цифр.

2. Дано натуральное четырехзначное число *п.* Выяснить, входит ли цифра 5 в запись этого числа.

Вариант 31.

1. Найти все двухзначные числа, которые при умножении на 2 заканчиваются цифрой 8, а при умножении на 3 – 4.

2. Даны n положительных и отрицательных чисел (все числа не равны нулю). Найти сумму отрицательных чисел и среднее геометрическое положительных чисел. Вывести количество отрицательных и количество положительных чисел.

Вариант 32.

1. Найти сумму всех чисел из промежутка от a до b, кратных 13 и 5.

2. Вводится целое число. Определить число нечетных цифр в данном числе.