

# Лабораторная работа № 3.

## Программирование циклических алгоритмов

**Цель работы:** Формирование умений и навыков решения задач на составление циклических алгоритмов и программ на языке C#.

### Теоретическая часть

#### Операторы цикла

Оператор цикла является важнейшим оператором и имеется в большинстве современных языков программирования.

**Циклом** называется последовательность действий, повторяемая несколько раз. Количество шагов (проходов) цикла зависит от некоторого условия, истинность которого проверяется перед исполнением тела цикла (*цикл с предусловием*) или после его исполнения (*цикл с постусловием*).

Очень часто условие представляет собой простую проверку, попадает ли в заданный диапазон счетчик цикла. Например, цикл нужно выполнить строго 10 раз, тогда счетчик будет изменяться от 1 до 10. Или нужно пробежаться по всем студентам в списке: в этом случае на каждом шаге цикла счетчик будет равен порядковому номеру студента. Для удобства работы с подобными алгоритмами во многих языках программирования реализован третий вид цикла – *цикл с параметром*.

#### Цикл с параметром (цикл for)

**Цикл с параметром** в C# предоставляет механизм итерации, в котором определенное условие проверяется перед выполнением каждой итерации.

Цикл с параметром имеет следующий формат:

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

**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.

**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**. Любая из частей оператора for может быть опущена (но точки с запятой надо оставить на своих местах!).

**Пример.** Вывести на экран квадраты и кубы целых чисел от 1 до 8.

```
using System;
class Program
{
    static void Main()
    {
        int i, i2, i3;
        for (i = 1; i <= 8; i++)
        {
            i2 = i * i;
            i3 = i2 * i;
            Console.WriteLine("{0,6}{1,6}{2,6}", i, i2, i3);
        }
        Console.ReadLine();
    }
}
```

или

```
using System;
class Program
{
    static void Main()
    {
        for (int i = 1; i <= 8; i++)
            Console.WriteLine("{0,6}{1,6}{2,6}", i, i * i, i * i * i);
        Console.ReadLine();
    }
}
```

*Результат работы программы:*

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512

**Пример.** Программа для вывода на экран делителей натурального числа  $n$ .

```
using System;
class Program
{
    static void Main()
    {
        int n, i;
        Console.Write("Введите натуральное число: ");
        n = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Делители числа:");
        for (i = 1; i <= n; i++)
        {
            if (n % i == 0)
                Console.WriteLine(i);
        }
        Console.ReadLine();
    }
}
```

```
}
```

*Результат работы программы:*

Введите натуральное число: 12

Делители числа:

```
1
2
3
4
6
12
```

## Цикл с предусловием (цикл while)

**Циклом с предусловием** называется такой цикл, в котором истинность условия проверяется каждый раз перед выполнением операторов тела цикла.

Формат оператора:

```
while ( выражение )
    оператор;
```

**Выражение** должно быть логического типа. Если результат вычисления выражения равен true, выполняется *простой* или *составной оператор*. Эти действия повторяются до тех пор, пока результатом выражения не станет значение false.

**Выражение** вычисляется перед каждой итерацией *цикла*. Если при первой проверке *выражение* равно false, цикл не выполнится ни разу.

**Пример.** Программа для подсчёта количества и суммы цифр во введённом натуральном числе.

```
using System;
class Program
{
    static void Main()
    {
        // n – вводимое число, k – количество цифр, S – сумма цифр
        int n, p, k = 0, S = 0;
        Console.Write("Введите натуральное число: ");
        n = Convert.ToInt32(Console.ReadLine());
        while (n != 0)
        {
            p = n % 10;
            S += p;
            k++;
            n /= 10;
        }
        Console.WriteLine("Количество цифр равно {0}", k);
        Console.WriteLine("Сумма цифр равна {0}", S);
        Console.ReadLine();
    }
}
```

*Результат работы программы:*

Введите натуральное число: 246

Количество цифр равно 3

Сумма цифр равна 12

## Цикл с постусловием (цикл do ... while)

**Циклом с постусловием** называется цикл, в котором истинность условия проверяется каждый раз после выполнения тела цикла.

Формат оператора:

```
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.

**Пример.** Программа для подсчета нечетных чисел. Командой выхода будет ввод нуля.

```
using System;
class Program
{
    static void Main()
    {
        int x, count = 0;
        Console.WriteLine("Вводите целые числа, а я буду считать нечетные");
        Console.WriteLine("Выход - по вводу нуля");
        do
        {
            x = Convert.ToInt32(Console.ReadLine());
            if (x % 2 != 0)
                count++;
        }
        while (x != 0);
        Console.WriteLine("Нечетных чисел " + count);
        Console.ReadLine();
    }
}
```

*Результат работы программы:*

```
Вводите целые числа, а я буду считать нечетные
Выход - по вводу нуля
5
6
7
9
0
Нечетных чисел 3
```

## Вложенные циклы

Циклы можно вкладывать друг в друга, при этом на разных уровнях могут использоваться циклы разных типов. Для вывода таблиц, обработки значений функции двух переменных используются двойные циклы.

Циклы закрываются по "принципу матрешки": первый цикл закрывается последним. Вот так:

```
while (...)
{
    // while-начало
    for (...)
    {
        // for-начало
    }
    // for-конец
}
// while-конец
```

**Пример.** Вывести на экран таблицу чисел

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

Как поступить, если проделать данную работу вручную? Очевидно, сначала следует написать в ряд 5 целых чисел (это внутренний цикл) и повторить эту операцию 4 раза (это внешний цикл).

```
using System;
class Program
{
    static void Main()
    {
        int i, j;
        for (i = 1; i <= 4; i++) // меняется по строкам
        {
            for (j = 1; j <= 5; j++) // меняется по столбцам
                Console.Write("{0,3}", j); // вывод нескольких столбцов
            Console.WriteLine(); // переход на следующую строку
        }
        Console.ReadLine();
    }
}
```

**Пример.** Вывести на экран треугольную таблицу, состоящую из чисел:

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

В такой таблице число столбцов равно номеру строки. Так в первой строке – 1 столбец, во второй – 2 и т.д. Поэтому текст программы будет таким:

```
using System;
class Program
{
    static void Main()
    {
        int i, j;
        for (i = 1; i <= 5; i++)
        {
            for (j = 1; j <= i; j++)
                Console.Write("{0,3}", i) ;
            Console.WriteLine();
        }
    }
}
```

```

    }
    Console.ReadLine();
}
}

```

## Оператор прерывания блока break

Оператор `break` можно использовать, чтобы прервать текущий блок цикла `while`, `do`, `for` или выполнение `case` в составе `switch`. Вот, например, фрагмент программы:

```

int a = 1;
while (a < 10)
{
    Console.Write(" " + a);
    a++;
}

```

При его выполнении на экран будет выведена строка: 1 2 3 4 5 6 7 8 9

Теперь добавим оператор `break`, который прервет выполнение цикла, как только `a` будет без остатка делиться на 5:

```

int a = 1;
while (a < 10)
{
    if (a % 5 == 0) break;
    Console.Write(" " + a);
    a++;
}

```

Теперь на экран выведется строка: 1 2 3 4

Если оператор `break` применяется в целом ряде вложенных циклов, то он прерывает выполнение только самого внутреннего цикла.

В отношении оператора `break` необходимо также иметь в виду следующее. Во-первых, в теле цикла может присутствовать несколько операторов `break`, но применять их следует очень аккуратно, поскольку чрезмерное количество операторов `break` обычно приводит к нарушению нормальной структуры кода. И во-вторых, оператор `break`, выполняющий выход из оператора `switch`, оказывает воздействие только на этот оператор, но не на объемлющие его циклы.

## Оператор завершения итерации цикла continue

С помощью оператора `continue` можно организовать преждевременное завершение шага итерации цикла в обход обычной структуры управления циклом. Оператор `continue` осуществляет принудительный переход к следующему шагу цикла, пропуская любой код, оставшийся невыполненным. Таким образом, оператор `continue` служит своего рода дополнением оператора `break`.

В циклах `while` и `do-while` оператор `continue` вызывает передачу управления непосредственно условному выражению, после чего продолжается процесс выполнения цикла. А в цикле `for` сначала вычисляется итерационное выражение, затем условное выражение, после чего цикл продолжается:

**Пример.**

```
using System;
```

```

class Program
{
    static void Main()
    {
        int y;
        Console.WriteLine("Мы будем вычислять значение функции  $y=2*x+5$ ");
        Console.WriteLine("в промежутках [1,6] и [13,18]");
        for (int x = 1; x <= 18; x++)
        {
            if (x > 6 && x < 13) continue;
            y = 2 * x + 5;
            Console.WriteLine("x= {0} y= {1}", x, y);
        }
        Console.ReadLine();
    }
}

```

## Практическая часть

**Пример 1.** Составить программу для вывода таблицы значений функции  $y = x^2 + 1$  на отрезке  $[a; b]$  с шагом  $h = 1$ .

```

using System;
class Program
{
    static void Main()
    {
        double a, b, x;
        Console.WriteLine("Введите концы отрезка:");
        a = Convert.ToDouble(Console.ReadLine());
        b = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("|-----|");
        Console.WriteLine("|      X      |      Y      |");
        Console.WriteLine("|-----|");
        for (x = a; x <= b; x++)
            Console.WriteLine("|{0,5} |{1,5} |", x, x * x + 1);
        Console.WriteLine("|-----|");
        Console.ReadLine();
    }
}

```

*Результат работы программы:*

Введите концы отрезка:

1  
10

-----	
X	Y
-----	
1	2
2	5
3	10
4	17
5	26
6	37
7	50
8	65

9	82
10	101
-----	

**Пример 2.** Составить программу вычисления факториала натурального числа.

Факториал (!) числа есть произведение последовательности целых чисел, например  $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$ . При накапливании произведения следует помнить, что начальным значением для переменной будет не нуль (иначе произведение всегда будет равно нулю), а единица.

```
using System;
class Program
{
    static void Main()
    {
        int n;
        long f = 1;
        Console.WriteLine("Введите натуральное число n <=20");
        n = Convert.ToInt32(Console.ReadLine());
        for (int i = 2; i <= n; i++)
            f *= i;
        Console.WriteLine("{0}! = {1} ", n, f);
        Console.ReadLine();
    }
}
```

*Результат работы программы:*

```
Введите натуральное число n <=20
5
5! = 120
```

**Пример 3.** Вычислить сумму конечного ряда  $S = \sum_{k=1}^n \frac{k}{3k^2 + 5}$ , т.е. сумму первых  $n$  членов последовательности  $\frac{k}{3k^2 + 5}$  ( $k = 1, 2, 3, \dots, n$ ).

```
using System;
class Program
{
    static void Main()
    {
        int n;
        double S = 0;
        Console.WriteLine("Введите количество членов последовательности");
        n = Convert.ToInt32(Console.ReadLine());
        for (int k = 1; k <= n; k++)
            S = S + (double)k / (3 * k * k + 5);
        Console.WriteLine("Сумма ряда S = " + S);
        Console.ReadLine();
    }
}
```

*Результат работы программы:*

```
Введите количество членов последовательности
10
Сумма ряда S = 0,683205078530812
```



12,1824938304434

**Пример 4.** Найти сумму ряда  $S = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2} + \dots$ , пока члены ряда больше  $eps > 0$ .

```
using System;
class Program
{
    static void Main()
    {
        int n = 1;
        double S = 0, eps;
        Console.WriteLine("Введите eps > 0");
        eps = Convert.ToDouble(Console.ReadLine());
        while (1.0 / (n * n) > eps)
        {
            S = S + 1.0 / (n * n);
            n++;
        }
        Console.WriteLine("Сумма ряда S = " + S);
        Console.ReadLine();
    }
}
```

*Результат работы программы:*

```
Введите eps > 0
0,001
Сумма ряда S = 1,61319070032792
Введите eps > 0
0,0001
Сумма ряда S = 1,63488390018489
```

**Пример 5.** Вывести на экран все трехзначные числа, сумма цифр которых равна заданному натуральному числу  $n$  ( $1 \leq n \leq 27$ ).

*1-й способ.* С использованием одного цикла.

```
using System;
class Program
{
    static void Main()
    {
        int n, x, a, b, c;    // n - введенное число, x - трехзначное число
        Console.WriteLine("Введите натуральное n <= 27");
        n = Convert.ToInt32(Console.ReadLine());
        for (x = 100; x <= 999; x++)
        {
            a = x / 100;      // a - число сотен
            b = x / 10 % 10;   // b - число десятков
            c = x % 10;        // c - число единиц
            if (a + b + c == n)
                Console.Write("{0,5}", x);
        }
        Console.ReadLine();
    }
}
```

**2-й способ.** С использованием вложенных циклов.

```
using System;
class Program
{
    static void Main()
    {
        int n, a, b, c;
        Console.WriteLine("Введите натуральное n <=27");
        n = Convert.ToInt32(Console.ReadLine());
        for (a = 1; a <= 9; a++)
            for (b = 0; b <= 9; b++)
                for (c = 0; c <= 9; c++)
                    if (a + b + c == n)
                        Console.Write("{0,5}", a * 100 + b * 10 + c);
        Console.ReadLine();
    }
}
```

## Задания для самостоятельной работы

1. Используя цикл с параметром, составьте программы для построения таблицы соответствия.

Например, для задания 1.1. таблица должна иметь вид:

км	версты
1	0,9374
2	1,8748
3	2,8121
4	3,7495
5	4,6869
6	5,6243
7	6,5617
8	7,4991
9	8,4364
10	9,3738
11	10,3112
12	11,2486
13	12,1860
14	13,1234
15	14,0607
16	14,9981
17	15,9355
18	16,8729
19	17,8103
20	18,7477

1.1. Составить программу для построения таблицы соответствия между расстоянием в километрах и расстоянием в верстах для значений 1, 2, ..., 20 километров<sup>1</sup>.

1.2. Составить программу для построения таблицы соответствия между весом в фунтах и весом в килограммах для значений 1, 2, ..., 20 фунтов<sup>2</sup>.

1.3. Составить программу для построения таблицы соответствия между длиной в метрах и длиной в саженьях и аршинах для значений 1, 2, ..., 20 метров<sup>1</sup>.

1.4. Составить программу для построения таблицы соответствия расстояний в дюймах расстояниям в сантиметрах для значений 1, 2, ..., 20 дюймов<sup>3</sup>.

<sup>1</sup> Верста = 500 сажень; 1 сажень = 3 аршина; 1 аршин = 16 вершков; 1 вершок = 44,45 мм.

<sup>2</sup> 1 фунт = 453 г

1.5. Составить программу для построения таблицы соответствия расстояний в километрах расстояниям в милях для значений 1, 2, ..., 20 километров<sup>4</sup>.

1.6. Одна штука некоторого товара стоит  $x$  руб. Составить программу для построения таблицы стоимости 1, 2, 3, ..., 20 штук этого товара.

1.7. Составить программу для построения таблицы соответствия между расстоянием в милях и расстоянием в километрах и метрах для значений 1, 2, ..., 20 миль<sup>4</sup>.

1.8. Составить программу для построения таблицы соответствия между температурными шкалами Цельсия и Фаренгейта в интервале температур от точки замерзания воды до точки ее кипения ( $T_F = 9/5 * T_C + 32$ ) с дискретностью в один градус.

1.9. Составить программу для построения таблицы перевода 1, 2, ..., 20 долларов США в рубли по текущему курсу (значение курса вводится с клавиатуры).

1.10. Составить программу для построения таблицы значений  $x$  и  $y$ , если  $x$  изменяется по формуле  $x = n^2 + 2,5$ , а  $y = 2x - n$ , для  $n$  в диапазоне  $[0; 20]$ . В колонках должны быть  $n$ ,  $x$ ,  $y$ .

## **2. Составить программы для обработки данных во время ввода.**

2.1. Известен рост каждого ученика двух классов. Определить средний рост учеников каждого класса.

2.2. Известны результаты двух спортсменов-пятиборцев в каждом из пяти видов спорта в баллах. Определить, какой спортсмен набрал наибольшую сумму баллов.

2.3. В области  $n$  районов. Известны количество жителей каждого района (в тыс. чел.) и плотность населения в нем (тыс. чел./ км<sup>2</sup>). Определить общую площадь территории области.

2.4. Известны оценки за контрольную работу по математике каждого из  $n$  учащихся. Определить, сколько «5», «4», «3» и «2» было выставлено в классе.

2.5. В области  $n$  районов. Известны количество жителей (в тыс. чел.) и площадь (в км<sup>2</sup>) каждого района. Определить среднюю плотность населения по области в целом.

2.6. В области  $n$  районов. Известно количество жителей каждого района (в тыс. чел.). Определить район, в котором проживает наибольшее количество жителей.

2.7. Известны оценки ученика по 10 предметам. Определить среднюю и максимальную оценки.

2.8. Известна масса каждого предмета из некоторого набора предметов. Определить среднюю массу и номер предмета с наименьшей массой. Если таких предметов несколько, то должен быть найден последний из них.

2.9. Известны оценки за контрольную работу по информатике каждого из  $n$  учащихся. Определить количество «5» и средний балл всего класса.

2.10. Известна масса каждого из 10 предметов. Определить общую массу всего набора предметов и номер предмета с наибольшей массой. Если таких предметов несколько, то должен быть найден последний из них.

## **3. Используя цикл с условием, составить программы для решения следующих задач:**

3.1. Бизнесмен взял ссуду  $m$  тысяч рублей в банке под 20% годовых. Через сколько лет его долг превысит  $s$  тысяч рублей, если за это время он не будет отдавать долг?

---

<sup>3</sup> 1 дюйм = 25,4 мм

<sup>4</sup> 1 миля = 1,609 км

3.2. Урожай яблок в 2000 году составил  $X$  тонн. Далее каждый год урожай уменьшался на 20%. Начиная с какого года, будет собрано менее  $Y$  тонн?

3.3. Начав тренировки, лыжник в первый день пробежал  $d$  км. Каждый следующий день он увеличивал длину пробега на 10% от пробега предыдущего дня. Определить, в какой день он пробежал больше  $s$  км.

3.4. В некотором году (назовем его условно первым) площадь участка составляла  $X$  гектаров. После этого каждый год площадь участка увеличивалась на 5%. Определить, в каком году площадь участка станет больше  $Y$  гектаров.

3.5. Гражданин 1 числа некоторого месяца (назовем его условно первым) открыл счет в банке, вложив  $m$  руб. Через каждый месяц размер вклада увеличивается на 2% от имеющейся суммы. Определить, за какой месяц величина ежемесячного увеличения вклада превысит  $p$  руб.

3.6. Ежемесячная стипендия студента составляет  $A$  рублей. Расходы на проживание составили  $B$  рублей в первый месяц. Рост цен ежемесячно увеличивает расходы на 5%. Определить, через сколько месяцев студенту понадобится помощь родителей.

3.7. Гражданин 1 числа некоторого месяца открыл счет в банке, вложив  $m$  руб. Через каждый месяц размер вклада увеличивается на 2% от имеющейся суммы. Определить, через сколько месяцев размер вклада превысит  $s$  руб.

3.8. У школьника было  $S$  рублей. Бутылка с соком стоит  $P$  рублей. Пустая бутылка стоит  $B$  рублей. Вычислить, сколько бутылок сока может выпить школьник, если он сдает пустые бутылки и на вырученные деньги покупает новые.

3.9. Начав тренировки, лыжник в первый день пробежал  $d$  км. Каждый следующий день он увеличивал длину пробега на 10% от пробега предыдущего дня. Определить, в какой день суммарный пробег за все дни превысит  $s$  км.

3.10. В первый час туристы прошли 5 км. Каждый следующий час их скорость снижалась на 10%. Через сколько часов туристы преодолели более  $S$  км?

#### **4. Составить программы для решения следующих задач:**

4.1. Вывести на экран все трехзначные числа, не превосходящие заданного  $n$ , у которых есть одинаковые цифры. Подсчитать количество таких чисел.

4.2. Вывести на экран все двузначные числа, содержащие цифру 3 или 7. Подсчитать количество таких чисел.

4.3. Вывести на экран те трехзначные числа, не превосходящие заданного  $n$ , у которых все цифры разные. Подсчитать количество таких чисел.

4.4. Вывести на экран все двузначные числа, оканчивающиеся на 2 или 5. Подсчитать количество таких чисел.

4.5. Вывести на экран все трехзначные числа, не превосходящие заданного  $n$ , десятичная запись которых есть строго возрастающая последовательность цифр. Подсчитать количество таких чисел.

4.6. В трехзначном числе зачеркнули первую цифру слева. Когда полученное двузначное число умножили на 7, то получили исходное трехзначное число. Найти это число.

4.7. Вывести на экран все двузначные числа, делящиеся на каждую из своих цифр. Подсчитать количество таких чисел.

4.8. Вывести на экран все симметричные четырехзначные числа. Например, 7667, 1331. Подсчитать количество таких чисел.

4.9. Вывести на экран все четырехзначные числа, в которых ровно три одинаковые цифры. Подсчитать количество таких чисел.

4.10. Вывести на экран все трехзначные числа, не превосходящие заданного  $n$ , десятичная запись которых есть строго убывающая последовательность цифр. Подсчитать количество таких чисел.

**5. Используя вложенные циклы, составить программы для решения следующих задач:**

5.1. Дано натуральное число  $n$ . Можно ли представить в виде суммы квадратов трех натуральных чисел? Если можно, то вывести на экран такие тройки  $x$ ,  $y$  и  $z$  натуральных чисел, что  $x^2 + y^2 + z^2 = n$ . Решения, которые получаются перестановкой  $x$ ,  $y$  и  $z$ , считать совпадающими.

5.2. Найти все натуральные числа из промежутка от  $a$  до  $b$ , у которых сумма делителей кратна 3.

5.3. Заданное двузначное число  $n$  представить в виде суммы квадратов двух натуральных чисел или выдать сообщение, что это невозможно.

5.4. Найти все натуральные числа из промежутка от  $a$  до  $b$ , у которых количество делителей равно  $k$ .

5.5. Вывести на экран все натуральные решения уравнения  $n^2 + m^2 = k^2$ , где  $n$ ,  $m$  и  $k$  не превосходят 20. Решения, которые получаются перестановкой  $n$  и  $m$ , считать совпадающими.

5.6. Написать программу для нахождения всех прямоугольников, площадь которых равна заданному натуральному числу  $S$  и стороны выражены натуральными числами. Решения, которые получаются перестановкой размеров сторон, считать совпадающими.

5.7. Вывести на экран все совершенные числа из диапазона от 1 до 10000. *Совершенным* называется такое число, которое равно сумме всех своих делителей, за исключением самого числа, например:  $28=1+2+4+7+14$ .

5.8. Написать программу для нахождения всех прямоугольных параллелепипедов, объем которых равен заданному натуральному числу  $V$  и ребра выражены натуральными числами. Решения, которые получаются перестановкой размеров ребер параллелепипеда, считать совпадающими.

5.9. Найти количество делителей каждого из натуральных чисел от  $m$  до  $n$ .

5.10. Найти сумму делителей каждого из натуральных чисел от  $m$  до  $n$ .

**6. Используя вложенные циклы, составить программы для решения следующих задач:**

6.1. Дано натуральное число  $n$ . Найти натуральное число от 1 до  $n$  с максимальным количеством делителей. Если таких чисел несколько, то должно быть найдено максимальное из них.

6.2. Дано натуральное число  $n$ . Вывести все числа в интервале от 1 до  $n$ , у которых произведение всех цифр совпадает с суммой цифр данного числа.

6.3. Даны натуральные числа  $n$  и  $m$ . Получить все меньшие  $n$  натуральные числа, сумма цифр которых равна  $m$ . Если такие числа отсутствуют, то вывести соответствующее сообщение.

6.4. Среди натуральных чисел от  $m$  до  $n$  найти число с максимальной суммой цифр.

6.5. Дано натуральное число  $n$ . Получить все простые делители этого числа.

6.6. Дано натуральное число  $n$ . Составить программу вывода цифр, не входящих в десятичную запись числа  $n$  (в порядке возрастания). Например: в числе 357 нет цифр: 0, 1, 2, 4, 6, 8, 9.

6.7. Дано натуральное число  $n$ . Найти натуральное число от 1 до  $n$  с максимальной суммой делителей. Если таких чисел несколько, то должно быть найдено минимальное из них.

6.8. Найти цифровой корень целого числа  $N$ . Цифровой корень находится суммой через сумму цифр числа до тех пор, пока эта сумма не станет цифрой. Например: цифровой корень числа 34697 равняется 2, т.к.  $3+4+6+9+7=29$ ,  $2+9=11$ ,  $1+1=2$ .

6.9. В данном натуральном числе переставить цифры таким образом, чтобы образовалось наибольшее число, записанное этими же цифрами.

6.10. В данном натуральном числе переставить цифры таким образом, чтобы образовалось наименьшее число, записанное этими же цифрами.

**7. Разработайте приложения для решения задач из сборника задач по программированию согласно вашему варианту (В.И. Великодный. Задачи по программированию: Учебное пособие. Раздел «Циклы»).**

№ варианта	Задания из лаб. работы	Задания из сборника задач по программированию
1	1.1, 2.1, 3.1, 4.1, 5.1, 6.1	3.1-1, 3.2-1, 3.3-1, 3.4-1
2	1.2, 2.2, 3.2, 4.2, 5.2, 6.2	3.1-2, 3.2-2, 3.3-2, 3.4-2
3	1.3, 2.3, 3.3, 4.3, 5.3, 6.3	3.1-3, 3.2-3, 3.3-3, 3.4-3
4	1.4, 2.4, 3.4, 4.4, 5.4, 6.4	3.1-4, 3.2-4, 3.3-4, 3.4-4
5	1.5, 2.5, 3.5, 4.5, 5.5, 6.5	3.1-5, 3.2-5, 3.3-5, 3.4-5
6	1.6, 2.6, 3.6, 4.6, 5.6, 6.6	3.1-6, 3.2-6, 3.3-6, 3.4-6
7	1.7, 2.7, 3.7, 4.7, 5.7, 6.7	3.1-7, 3.2-7, 3.3-7, 3.4-7
8	1.8, 2.8, 3.8, 4.8, 5.8, 6.8	3.1-8, 3.2-8, 3.3-8, 3.4-8
9	1.9, 2.9, 3.9, 4.9, 5.9, 6.9	3.1-9, 3.2-9, 3.3-9, 3.4-9
10	1.10, 2.10, 3.10, 4.10, 5.10, 6.10	3.1-0, 3.2-0, 3.3-0, 3.4-0