Л а б о р а т о р н а я р а б о т а № 6

**ЗАДАЧИ С ПОСЛЕДОВАТЕЛЬНОСТЯМИ**

**Цель работы:**

* Получение навыков в использовании множеств и организации ввода и вывода данных множества.
* Получение практических навыков в выполнении операций над множествами

**Краткие теоретические сведения.**

**Циклы**

Циклы в языках программирования предназначены для построения конструкции, выполняющей заданный блок кода некоторое количество раз, которое определяется тем или иным условием. *C*# предоставляет четыре разных варианта построения циклов.

**Цикл *for***

Цикл for имеет следующий синтаксис:

**for**(инициализация счетчика; условие; итератор)

{

// оператор (набор операторов)

}

*Инициализатор* – это выражение, вычисляемое перед выполнением тела цикла. Обычно здесь инициализируется локальная переменная, которая будет использоваться как счетчик.

*Условие* – это выражение, которое проверяется перед каждой новой итерацией цикла. Если значение выражения равно *true*, то будет выполнено тело цикла, если *false* – произойдет выход из цикла и выполнится следующая за ним инструкция.

*Итератор* – это выражение, вычисляемое после каждой итерации. Обычно здесь происходит изменение локальной переменной, объявленной в инициализаторе.

Перечисленные выражения: *инициализатор*, *условие* и *итератор* разделяются символом “точка с запятой”. Цикл *for* удобно использовать, когда известно количество повторений. Пример работы с циклом *for*:

**for**(**int**i = 0; i<3; i++){

console.writeline(“Квадратчислаi ” + i\*i);

}

В нем мы инициализируем локальную переменную значением 0, затем проверяем, что переменная меньше трех, выполняем тело цикла – выводим в консоль значение числа *i* в квадрате. На каждой итерации прибавляем к переменной *i* значение 1, снова проверяем условие и выполняем тело цикла, и так, до тех пор, пока условие *(i< 3)*будет истинным.

**Циклы *while* и *do/while***

В *C*# цикл *while* имеет следующую конструкцию:

**while**(условие)

{

// Тело цикла

}

В этом цикле проверяется условие, и если оно истинно, то выполняется набор операторов внутри тела цикла. Обязательно убедитесь, что изменяете переменную, от которой зависит условие, иначе цикл станет бесконечным, если, конечно, это не является целью. Примерработысциклом *while*:

**int**i = 0;

**int** number = 3;

**while**(i<= number){

Console.WriteLine("Итерацияцикланомер " + i);

i++;

}

Console.ReadKey();

Цикл *do/while*имеетследующийсинтаксис:

**do**{

//Тело цикла

}**while**(условие);

Отличие *do/while* заключается в том, что проверка условия происходит после тела цикла, что приводит к тому, что вне зависимости от условия цикл выполнится хотя бы один раз.

Пример работы с циклом *do/while*:

**int** j = 0;

**int** number2 = 3;

**do**

{

Console.WriteLine("Итерацияцикланомер " + j);

j++;

}**while**(j > number2);

Console.ReadKey();

**Цикл *foreach***

Последний вариант цикла, который мы рассмотрим в этом уроке – это *foreach*. Синтаксис оператора *foreach* имеет следующий вид:

**foreach**(тип имя\_переменной\_циклаin коллекция)

{

// Тело цикла

}

Оператор *foreach* используется для обхода коллекций, последовательно переходя от элемента к элементу в цикле. В данном случае, под коллекцией понимается тип, который:

* реализуетинтерфейс    
  *System.Collections.IEnumerable* или *System.Collections.Generic.IEnumerable<T>;*
* реализует открытый метод *GetEnumerator*, возвращающий интерфейс, класс или структуру, имеющие открытое свойство *Current* и метод *MoveNext*.

Пример работы с оператором *foreach*:

**int**[]nums = {6, 3, 6, 8, 9, 12, 4, 5, 88, 54, 3, 66, 78, 10, 12, 5, 7, 9, 3, 5};

**int** result = 0;

**foreach**(**int** n innums)

{

**if**(n >10)

{

result++;

}

}

Console.WriteLine($"Количествочиселвмассивебольше 10: {result}");

В нем мы определяем количество чисел, которые больше десяти, в исходном массиве.

Тип переменной цикла в операторе *foreach* можно задавать явно, как это было сделано в примере выше, так и не явно с помощью ключевого слова *var*:

result = 0;

**foreach**(**var** n innums)

{

**if**(n <10)

{

result++;

}

}

Console.WriteLine($"Количествочиселвмассивеменьше 10: {result}");

**Операторы перехода**

Язык *C*# предоставляет специальные операторы для прерывания выполнения всего цикла и для принудительного завершения текущей итерации с переходом к следующей. Первую задачу решает оператор *break*. Если в программе используется несколько вложенных циклов, то при использованиии *break*, выход будет выполнен только из того цикла, где этот оператор был вызван. Оператор *continue* принудительно завершает текущую итерацию цикла и переходит к следующей. При этом для цикла *while* и *do/while* происходит переход к условному выражению, а в цикле *for* сначала  вычисляется итерационное выражение, а затем проверяется условие:

Console.WriteLine("### Операторыперехода");

**for**(**int**i = 0; i<nums.Length; i++)

{

**if**(nums[i]>10)

{

Console.WriteLine($"{nums[i]}> 10");

**continue**;

}

**if**(i>7)

{

Console.WriteLine("Break cycle");

**break**;

}

}

***LINQ* как инструмент обхода коллекций**

Существует более удобный инструмент для работы с коллекциями – это язык запросов *LINQ*. Этой технологии будет посвящен отдельный урок, здесь же мы разберем несколько примеров работы с *LINQ* для того, чтобы показать некоторые из его возможностей и заинтересовать вас на дальнейшее изучение этого инструмента. Вначале нами будут рассмотрены некоторые из методов расширения последовательностей, которые предоставляет пространство имен *System.Linq*, после этого будет приведен пример, демонстрирующий работу с языком *LINQ*.

Ниже представлены два примера, первый – это вариант реализации некоторого алгоритма с использованием циклов, второй – этот же алгоритм, но решение построено с помощью *LINQ*.

**Пример без *LINQ***

Обратимся к ранее созданному массиву *nums*, извлечем из него только четные элементы и возведем их в квадрат, полученные значения поместим в новый массив *evenSq*:

**var**evenSq = new List<**int**>();

**foreach**(**var** n innums)

{

**if**(n % 2 == 0)

{

evenSq.Add(n \* n);

}

}

**Пример с *LINQ***

Выполним туже операцию с помощью *LINQ*:

evenSq = nums

.Where(v => v % 2 == 0)

.Select(v => v \* v)

.ToList();

С помощью методов расширения из *System.Linq*можно выполнять над коллекциями операции фильтрации, преобразования, группировки, агрегирования, формирования коллекций и т.д. Рассмотрим некоторые из этих возможностей.

**Фильтрация**

Для фильтрации элементов в коллекции используйте метод *Where*, которому в качестве аргумента передается предикат, если его значение истинно для элемента коллекции, то он (элемент) остается, в противном случае отбрасывается.

Извлечем из массива *nums* числа, значение которых больше 10:

**var** arr1 = nums.Where(v => v >10).ToList(); // Take elements thats higher then 10

**Преобразование**

Метод *Select* применяет переданную ей функцию к элементам коллекции и формирует на базе полученных значений новую коллекцию. Вычтемизэлементов *nums* константу 7:

**var** arr2 = nums.Select(v => v - 7).ToList(); // Sub 7 from every elements

Построим на базе *nums* новый массив, элементами которого будут *bool* значения: *true*, если соответствующий элемент в *nums* четный, *false* – в ином случае.

**var** arr3 = nums.Select(v => v % 2 == 0).ToList();

Рассмотрим ещё один пример работы с *LINQ*.

Создадим класс *FootballTeams* для представления футбольной команды:

**publicclass**FootballTeam

{

**public** string Name {**get**; **set**; }

**public** string Country {**get**; **set**; }

**publicint** Group {**get**; **set**; }

**publicint**PlaceGroup{**get**; **set**; }

}

Создадим метод для генерации данных о результатах футбольных матчей:

**privatestatic** List<FootballTeam>CreateList()

{

**return**new List<FootballTeam>

{

{newFootballTeam(){ Name="Zenit", Country="Russia", Group= 1, PlaceGroup= 3, NumberpointsScored = 3}},

{newFootballTeam(){ Name="Ajax", Country="Holand", Group= 1, PlaceGroup= 2, NumberpointsScored = 4}},

{newFootballTeam(){ Name="Manchester United", Country="England", Group= 1, PlaceGroup= 1, NumberpointsScored = 6}},

{newFootballTeam(){ Name="Bavaria", Country="Germany", Group= 2, PlaceGroup= 1, NumberpointsScored = 8}},

{newFootballTeam(){ Name="Spartak", Country="Russia", Group= 2, PlaceGroup= 2, NumberpointsScored= 6}},

{newFootballTeam(){ Name="Real", Country="Italy", Group= 2, PlaceGroup= 3, NumberpointsScored = 3}},

{newFootballTeam(){ Name="Arsenal", Country="England", Group= 3, PlaceGroup= 2, NumberpointsScored = 9}},

{newFootballTeam(){ Name="Shakter", Country="Ukrane", Group=3, PlaceGroup= 3, NumberpointsScored = 6}},

{newFootballTeam(){ Name="Barselona", Country="Espane", Group= 3, PlaceGroup= 1, NumberpointsScored = 12}}

};

}

Создадим переменную для хранения результатов матчей:

**var**teams = CreateList();

Воспользуемся *LINQ* запросом и выберем из нашей коллекции все команды, которые занимают первое место в группах и выведем на экран название команды и количество заработанных очков.

**var** teams = CreateList();

**var**selectedTeams = **from** team in teams

**where**team.PlaceGroup == 1 // фильтрация

**orderby**team.Name // упорядочиваниепоимени

**select** team; // выборобъекта

**foreach**(**var** team inselectedTeams)

{

Console.WriteLine(team.Name + " " + team.NumberpointsScored);

}

Благодаря *LINQ* можно применять один и тот же запрос к разным источникам данных, более того, такие запросы экономят ресурсы и выполняются гораздо быстрее, чем обычный обход коллекций средствами *foreach*, так как работа идет с уже отфильтрованными данными и в этом случае не нужно перебирать всю коллекцию.

**Практическая часть.**

**Варианты**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Вариант** | **Задание 1** | **Задание 2** | **Задание 3** | **Задание 4** | **Задание 5** |
|  | **1** | **13** | **5** | **3** | **1** |
|  | **2** | **12** | **6** | 4 | 2 |
|  | **3** | **11** | **7** | **5** | **3** |
|  | **4** | **10** | **8** | **6** | **4** |
|  | **5** | **9** | **9** | **7** | **5** |
|  | **6** | **8** | **10** | **1** | **6** |
|  | **7** | **7** | **11** | **2** | **7** |
|  | **8** | **6** | **12** | **3** | **8** |
|  | **9** | **5** | **13** | **4** | **9** |
|  | **10** | **4** | **1** | **5** | **10** |
|  | **11** | **3** | **2** | **6** | **11** |
|  | **12** | **2** | **3** | **7** | **12** |
|  | **13** | **1** | **4** | **8** | **13** |
|  | **14** | **13** | **5** | **9** | **1** |
|  | **15** | **12** | **6** | **10** | **2** |
|  | **16** | **11** | **7** | **11** | **3** |
|  | **17** | **10** | **8** | **12** | **4** |
|  | **18** | **9** | **9** | **1** | **5** |
|  | **19** | **8** | **10** | **2** | **6** |
|  | **20** | **7** | **11** | **3** | **7** |

***Все задания консольных приложений выполнить в своих классах. Запускать из единого меню.***

*Задание 1. Напишите консольное приложение, выполняющее вычисления в процессе ввода значений.*

1. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество отрицательных чисел и максимальное количество подряд идущих отрицательных чисел.
2. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество четных чисел и минимальное количество подряд идущих отрицательных чисел.
3. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество чисел кратных 3 и количество групп подряд идущих чисел кратных 5.
4. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество чисел кратных 5 и минимальное положительное число.
5. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество нечетных чисел и максимальное отрицательное число.
6. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество четных чисел и наиболее близкие соседние числа.
7. Дана последовательность чисел, заканчивающаяся нулем. Вычислить количество положительных чисел и наиболее далекие соседние числа.
8. Дана последовательность из вещественных чисел, начинающаяся с отрицательного числа и заканчивающаяся нулём. Определить какое число отрицательных чисел записано в начале последовательности.
9. Дана непустая последовательность неотрицательных целых чисел, заканчивающаяся нулем. Найти среднее арифметическое чисел этой последовательности.
10. Дана последовательность из вещественных чисел, начинающаяся с нечётного числа и заканчивающаяся нулём. Найти сумму всех идущих подряд в начале последовательности нечётных чисел.
11. Дана последовательности вещественных чисел, упорядоченная по возрастанию, но заканчивающаяся нулём. И число *n* не равное ни одному элементу последовательности. Найти сумму чисел последовательности меньших *n*. (Ввод начинать с *n*)
12. Дана последовательности вещественных чисел, упорядоченная по возрастанию, но заканчивающаяся нулём. И число n не равное ни одному элементу последовательности. Найти сумму чисел последовательности больших n. (Ввод начинать с *n*)
13. Дана последовательности вещественных чисел, заканчивающаяся нулём. И число *n* не равное ни одному элементу последовательности. Найти все пары элементов последовательности в интервале между которыми находится *n*. (Ввод начинать с *n*)
14. Дана последовательности вещественных чисел, заканчивающаяся нулём. И число *n* не равное ни одному элементу последовательности. Найти номера всех пар элементов последовательности в интервале между которыми находится *n*. (Ввод начинать с *n*)
15. Дана последовательности вещественных чисел, заканчивающаяся нулём. Указать расстояния между самыми далёкими числами.
16. Дана последовательности вещественных чисел, заканчивающаяся нулём. Указать расстояния между самыми близкими числами.
17. Дана последовательность из вещественных чисел, заканчивающаяся нулём. Найти произведение чисел, расположенных после минимального.
18. Дана последовательность из вещественных чисел, заканчивающаяся нулём. Найти сумму чисел, расположенных после максимального.
19. Дана последовательность из вещественных чисел, заканчивающаяся нулём. Найти произведение чисел, расположенных после максимального.
20. Дана последовательность из вещественных чисел, заканчивающаяся нулём. Найти сумму чисел, расположенных после минимального.

*Задание 2. Напишите консольное приложение, выполняющее обработку числовых последовательностей без использования массивов.*

1. Дана непустая последовательность положительных целых чисел *а1, а2, а3*, …, оканчивающаяся нулем. Получить *а1, а1∙ а2, а∙, а2∙ а3*, ..., 0.
2. Дано число *n*. Из чисел 1, 4, 9, 16, 25.... напечатать те, которые не пре­вышают *n*.
3. Среди чисел 1, 4, 9, 16, 25, ... найти первое число, большее *n*.
4. Дано число*n*.Напечатать тeнатуральные числа, квадрат которых не превышает *n*. 6
5. Дано число*n*. Найти первое натуральноe число, квадрат которого больше*n*.
6. Дано число *a (1< а ≤ 1,5).* Из чисел 1 +1/2, 1+1/3, ... напечатать те, которые не меньше *а*.
7. Дано число *a (1< а ≤ 1,5).* Среди чисел 1 +1/2 , 1+1/3, ... найти первое, меньшее *а*.
8. Рассмотрим последовательность чисел: 1 +1/2 , 1+1/3, ... 1+1/*n.* Напечатать все значения *n*, при которых все числа последовательности будут не меньше *a (1< а ≤ 1,5).*
9. Дано число *a (1< а ≤ 1,5).* Нaйти такое наименьшее *n*, что в последовательности чисел 1 +1/2, 1+1/3, ... 1+1/*n.* последнее число будет меньше *а*.
10. Дано вещественное число *а*. Из чисел 1 +1/2, 1+1/3, ..., напечатать те, которые меньше *а*.
11. Среди чисел 1, 1 +1/2, 1+1/2+1/3, ... найти первое, большее числа *п*.
12. Дано вещественное число *а*. Напечатать все значения *п*, при которых   
    1+1/2+1/3+ … +1/*n>a.*
13. Дано вещественное число *а*. Найти такое наименьшее значения *п*, при которых   
    1+1/2+1/3+ … +1/*n>a.*

*Задание 3. Напишите консольное приложение, используйте условный оператор в теле операторов цикла с условием. Во всех задачах использовать только одни оператор цикла.*

1. Дано натуральное число. Определить:
   1. количество шифр 3 в нем;
   2. сколько раз в нем встречается последняя цифра;
   3. количество четных цифр в нем. (Составное условие и более одного неполного условного оператора не использовать);
   4. сумму его цифр, больших пяти;
   5. произведение его цифр, больших семи;
   6. сколько раз в нем встречаются цифры 0 и 5 (всего).
2. Дано натуральное число. Определить:
   1. сколько раз в нем встречается цифра *а*;
   2. количество его цифр, кратных *z* (значение *z* вводится с клавиатуры*; z=2, 3, 4*);
   3. сумму его цифр, больших *a* (значение *а* вводится с клавиатуры; *0 ≤ а ≤ 8*);
   4. сколько раз в нем встречаются цифры *х* и *у*.
3. Дана непустая последовательность целых чисел, оканчивающаяся нулем. Найти:
4. сумму всех чисел последовательности, больших числа*х*;
5. количество всех четных чисел последовательности.
6. Дана последовательность ненулевых целых чисел, оканчивающая нулем. Определить, сколько раз в этой последовательности меняется знак. (Например, в последовательности 10, -4, 12, 56, -4 знак меняется 3 раза.)
7. Дано натуральное число.
8. Определить его максимальную цифру.
9. Определить его минимальную цифру.
10. Дано натуральное число.
11. Определить его максимальную и минимальную цифры.
12. Определить, на сколько его максимальная цифра превышает минимальную.
13. Найти сумму его максимальной и минимальной цифр.
14. Дало натуральное число, в котором все цифры различны. Определить порядковый номер его максимальной цифры, считая номера:
15. От конца числа;
16. от начала числа;
17. Дало натуральное число, в котором все цифры различны. Определить порядковый номер сто минимальной цифры, считая номера:
18. От конца числа;
19. От начала числа.
20. Дано натуральное число, в котором все шифры различны. Определить порядковые номера его максимальной и минимальной цифр, считая номера:
21. от конца числа;
22. от начала числа.
23. Найти 15 первых натуральных чисел, делящихся нацело на 19 и находящихся в интервале, левая граница которого равна 100.
24. Найти 20 первых натуральных чисел, делящихся нацело на 13 ни на 17 и находящихся в интервале, левая граница которого равна 500.
25. Найти 11 первых натуральных чисел, оканчивающихся на цифру 7, кратных числу 9 и находящихся в интервале, левая граница которого равна 100.
26. Дано натуральное число. Определить, сколько раз в нем встречается первая его цифра.

*Задание 4. Напишите оконное приложение. Используйте условный оператор после операторов цикла с условием.*

1. Дано натуральное число:
2. Верно ли, что сумма его цифр меньше *а*?
3. Верно ли, что произведение его цифр больше *b*?
4. Верно ли, что это число *k*-значное? Составное условие и вложенный условный оператор не использовать.
5. Верно ли, что его первая цифра превышает *m*?
6. Дано натуральное число:
7. Верно ли, что сумма его цифр больше *k*, а само число четное?
8. Верно ли, что количество его цифр есть четное число, а само число не превышает *b*?
9. Верно ли, что оно начинается цифрой *х* и заканчивается цифрой *у*?
10. Верно ли, что произведение его цифр меньше п. а само число делится на Ь?
11. Верно ли, что сумма его цифр больше *m*, а само число делится на *п*?
12. Дано натуральное число. Определить:
13. Есть ли в нем цифра 3;
14. есть ли в нем цифры 2 и 5.
15. Дано натуральное число.
16. Определить, есть ли в нем цифра *а*.
17. Верно ли, что в нем нет цифры *b*?
18. Верно ли, что цифра *а* встречается в нем более *k* раз?
19. Определить, есть ли в нем цифры *n* и *b*.
20. Дано натуральное число. Выяснить, является ли оно палиндромом ("перевертышем"), т. е. числом, десятичная запись которого читается одинаково слева направо и справа налево.
21. Дано натуральное число. Выяснить, является ли оно простым (простым называется натуральное число, больше 1, не имеющее других делителей, кроме единицы и самого себя). Оператор цикла с параметром не использовать.
22. Дано натуральное число. Установить, является ли последовательность его цифр при просмотре их справа налево упорядоченной по возрастанию. Например, для числа 5321 ответ положительный, для чисел 7820 и 9663— отрицательный и т. п.
23. Дано натуральное число. Установить, является ли последовательность его цифр при просмотре их справа налево упорядоченной по неубыванию. Например, для чисел 5321 и 9663 ответ положительный, для числа 7820 — отрицательный и т. п.
24. Дано натуральное число. Установить, является ли последовательность его цифр при просмотре их слева направо упорядоченной по возрастанию. Например, для числа 1478 ответ положительный, для чисел 1782 и 1668 — отрицательный и т. п.
25. Дано натуральное число. Установить, является ли последовательность его цифр при просмотре их слева направо упорядоченной по неубыванию. Например, для чисел 1368 и 1669 ответ положительный, для числа 1782 — отрицательный и т. п.
26. Дана последовательность вещественных чисел *a1, a2, a3, …, a15,* упорядоченная по возрастанию, и число *n*, не равное ни одному из чисел последовательности и такое, что *а <п< 15*. Найди элемент последовательности (его порядковый помер и значение), ближайший к *п*.
27. Дана последовательность вещественных чисел *a1, a2, a3, …, a15*. Определить, есть ли в последовательности отрицательные числа. В случае положительного ответа определить порядковый номер первого из них.

*Задание 5. Напишите консольное приложение.*

1. Дана непустая последовательность целых чисел, оканчивающаяся числом 100. Определить, есть ли в последовательности число 77? Если имеются несколько таких чисел, то определить порядковый номер первого из них.
2. Дана непустая последовательность целых чисел, оканчивающаяся числом 6.Определить, есть ли в последовательности хотя 6ы одно число, кратное шести? В случае положительного ответа определить порядковый номер первого из них.
3. Известны данные о количестве осадков, выпавших за каждый деньмарта. Верно ли, что осадков не было 10дней в месяц? Можно ли в программе использовать оператор цикла с условием?
4. Известны стоимости (в долларах) нескольких марок легковых автомобилей и мотоциклов. Верно ли, что средняя стоимость автомобилей превышает среднюю стоимость мотоциклов более чем в 3 раза?
5. Известны стоимости (в долларах) нескольких марок легковых автомобилей и мотоциклов. Верно ли, что стои­мость одного автомобиля превышает $5000, что больше стоимости любой марки мотоцикла?
6. Известен рост каждого ученика класса. Рост мальчиков условно задан отрицательными числами. Верно ли, что средний рост мальчиков пре­вышает средний рост девочек более чем на 10 см?
7. Даны натуральное число *п* и целые числа *a1, a2, a3, …, an*. Верно ли, что максимальное из чисел *аi*, превышает минимальное не более чем на 25.
8. Известна масса каждого человека из группы. Верно ли, что масса самого тяжелого из них превышает массу самого легкого более чем в 2 раза.
9. Даны натуральное число *п* и целые числа *х1. х:, ..., хn.* Какое число в последовательности чисел *хi*, встретится раньше: максимальное илы минимальное. Если максимальных или минимальных чисел несколько, то должны быть учтены самые первые из них.
10. Даны натуральное число *п* и целые числа *х1. х:, ..., х .* Какое число в последовательности чисел *хi*, встретится раньше: максимальное илы минимальное. Если максимальных или минимальных чисел несколько, то должны быть учтены самые последние из них.
11. Известен возраст группы людей в списке. Какой человек указан в спи­ске раньше: самый старший или самый молодой? (Должны учитывать­ся первые из людей одинакового возраста.)
12. Известны результаты (время в минутах), показанные автогонщиком участником соревнований "Формула-1" - на каждом этапе. Известно также, что на одном этапе он занял первое место и на одном - по­следнее. Верно ли, что этап, который он выиграл, был раньше этапа, на котором он занял последнее место?
13. Даны 20 чисел, образующие неубывающую последовательность, не­сколько чисел, идущие подряд, равны между собой. Найти количество таких чисел. Сколько различных чисел имеется в последовательности?

**Контрольные вопросы**

1. Как задать последовательность чисел?
2. Какие условные операторы цикла Вам известны?