Collections : Queue

- C# includes a Queue collection class in the System.Collection namespace.
- Queue stores the elements in FIFO style (First In First Out), exactly opposite of the Stack collection.
- It contains the elements in the order they were added.
- Queue collection allows multiple null and duplicate values.
- Use the Enqueue() method to add values and the Dequeue() method to retrieve the values from the Queue.

Property of Queue class

Count
Gets the number of elements contained in the Stack.

Methods of Queue

public virtual void Clear();
Removes all elements from the Queue.

public virtual bool Contains(object obj);
Determines whether an element is in the Queue.

public virtual object Dequeue();
Removes and returns the object at the beginning of the Queue.

public virtual void Enqueue(object obj);
Adds an object to the end of the Queue.

public virtual object[] ToArray();
Copies the Queue to a new array.

public virtual void TrimToSize();
Sets the capacity to the actual number of elements in the Queue.

Application Program which demonstrate use of Queue methods and properties.

```
using System;

using System.Collections;

public class Marvellous
{
        public static void Main(string[] args)
        {
                Queue qobj = new Queue();

                qobj.Enqueue('A');
                qobj.Enqueue('B');
                qobj.Enqueue('C');
                qobj.Enqueue('D');

                Console.WriteLine("Current queue: ");
                foreach (char c in qobj)
                {
                        Console.Write(c + " ");
                }

                qobj.Enqueue('F');
                qobj.Enqueue('G');

                Console.WriteLine("Current queue: ");
                foreach (char c in qobj)
                {
                        Console.Write(c + " ");
                }

                Console.WriteLine("Removing some values ");
                char ch = (char) qobj.Dequeue();

                Console.WriteLine("The removed value: {0}", ch);
                ch = (char) qobj.Dequeue();

                Console.WriteLine("The removed value {0},ch);
        }
   }
```