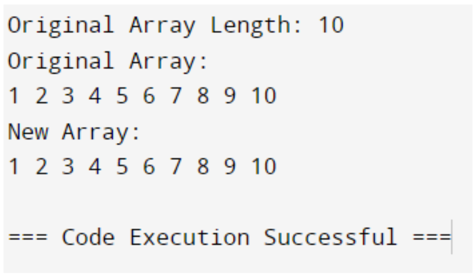
1. Use String for an immutable string, and use StringBuilder for a string to which frequent modifications are performed.
2. System.Array
3. Array.sort(array);
4. array.Length
5. Yes.
6. array1.CopyTo(array2, idx); copies the elements of array1 (starting from position idx) to array2 that already exists, and does not return any value; Clone() returns a shallow copy of the current array.

1.

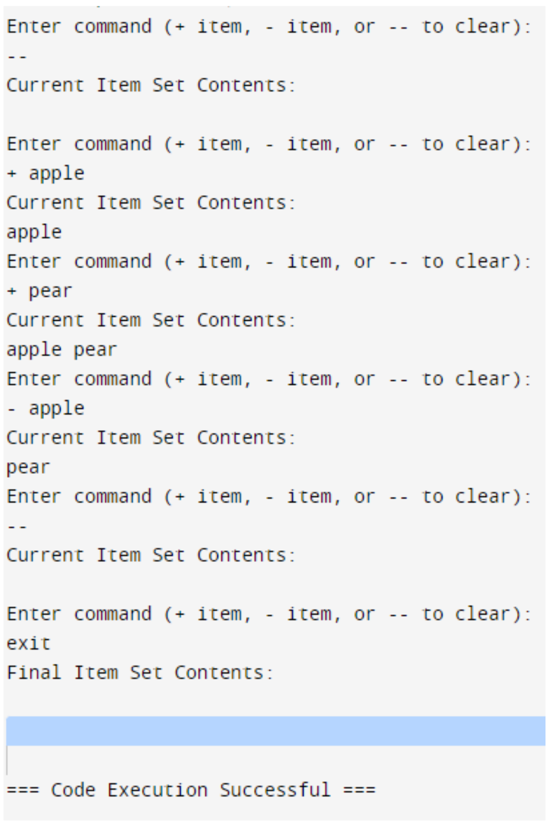
```
using System;

class Program
{
    static void Main(string[] args)
    {
        // Create the initial array of 10 integers from 1 to 10
        int[] originalArray = new int[10];
        for (int i = 0; i < 10; i++)
        {
            originalArray[i] = i + 1;
        }

        // Output the original array's Length property
        Console.WriteLine("Original Array Length: " + originalArray.Length);

        // Create a second array with the same length
        int[] newArray = new int[originalArray.Length];

        // Use a loop to read values from the original array and place them in the
new array
        for (int i = 0; i < originalArray.Length; i++)
        {
            newArray[i] = originalArray[i];
        }

        // Output both arrays
        Console.WriteLine("Original Array:");
        PrintArray(originalArray);
        Console.WriteLine("New Array:");
        PrintArray(newArray);
    }
    // Method to print an array
    static void PrintArray(int[] arr)
    {
        foreach (int num in arr)
        {
            Console.Write(num + “ “);
        }
        Console.WriteLine();
    }
}
```



```
Original Array Length: 10
Original Array:
1 2 3 4 5 6 7 8 9 10
New Array:
1 2 3 4 5 6 7 8 9 10

=== Code Execution Successful ===
```

2.

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main(string[] args)
    {
        // String set to store items
        HashSet<string> items = new HashSet<string>();

        while (true)
        {
            // Output the command prompt
            Console.WriteLine(“Enter command (+ item, - item, or -- to clear):”);

            // Read user input
            string userInput = Console.ReadLine();

            // Check user input
            if (userInput.StartsWith(“+ ”)) // Add item
            {
                string item = userInput.Substring(2); // Extract item name
                items.Add(item);
            }
            else if (userInput.StartsWith(“- “)) // Remove item
            {
                string item = userInput.Substring(2); // Extract item name
                items.Remove(item);
            }
            else if (userInput == “--”) // Clear items
            {
                items.Clear();
            }
            else if (userInput == “exit”) // Exit loop
            {
                // Output final item set contents
                Console.WriteLine(“Final Item Set Contents:”);
                foreach (string item in items)
                {
                    Console.WriteLine(item);
                }
                break; // Exit the loop
            }
            else
            {
                Console.WriteLine(“Invalid command.”); // Handle invalid
input
            }
        }
    }
}
```

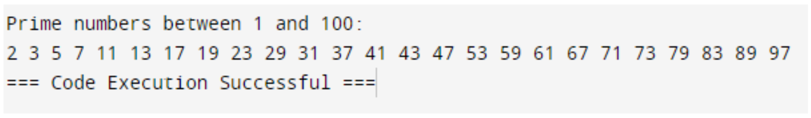


```
Enter command (+ item, - item, or -- to clear):
--
Current Item Set Contents:

Enter command (+ item, - item, or -- to clear):
+ apple
Current Item Set Contents:
apple
Enter command (+ item, - item, or -- to clear):
+ pear
Current Item Set Contents:
apple pear
Enter command (+ item, - item, or -- to clear):
- apple
Current Item Set Contents:
pear
Enter command (+ item, - item, or -- to clear):
--
Current Item Set Contents:

Enter command (+ item, - item, or -- to clear):
exit
Final Item Set Contents:



=== Code Execution Successful ===
```

3.

```
static int[] FindPrimeInRange(int startNum, int endNum)
        {
                List<int> primesList = new List<int>();

                bool IsPrime(int num)
                {
                        if (num <= 1)
                        {
                                return false;
                        }

                        for (int i = 2; i <= Math.Sqrt(num); i++)
                        {
                                if (num % i == 0)
                                {
                                        return false;
                                }
                        }

                        return true;
                }
```

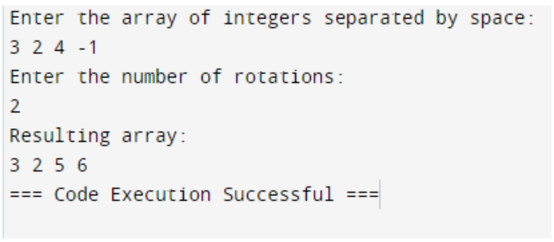
```
            for (int num = startNum; num <= endNum; num++)
            {
                if (IsPrime(num))
                {
                    primesList.Add(num);
                }
            }

            return primesList.ToArray();
        }
```



```
Prime numbers between 1 and 100:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
=== Code Execution Successful ===
```

4.

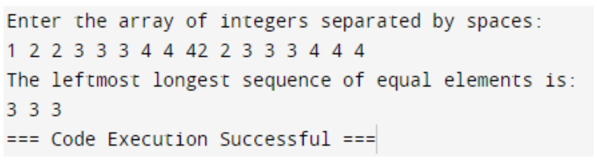
```
using System;

class Program
{
        static void Main(string[] args)
        {
                // Read the array of integers
                Console.WriteLine("Enter the array of integers separated by space:");
                string[] inputArray = Console.ReadLine().Split(' ');

                // Convert the input array to integers
                int[] nums = Array.ConvertAll(inputArray, int.Parse);

                // Read the number of rotations
                Console.WriteLine("Enter the number of rotations:");
                int k = int.Parse(Console.ReadLine());

                // Perform rotations and sum the obtained arrays position-wise
                int[] result = RotateAndSum(nums, k);

                // Output the result array
                Console.WriteLine("Resulting array:");
                foreach (int num in result)
                {
                        Console.Write(num + " ");
                }
        }

        static int[] RotateAndSum(int[] nums, int k)
        {
                int n = nums.Length;
                int[] rotatedArray = new int[n];
                int[] result = new int[n];

                for(int i = 1; i <= k; ++i){
                        for(int j = 0; j < n; ++j){
                                result[j] += nums[(j+n-i) % n];
                        }
                }

                return result;
        }

}
```



```
Enter the array of integers separated by space:
3 2 4 -1
Enter the number of rotations:
2
Resulting array:
3 2 5 6
=== Code Execution Successful ===
```

5.

```
using System;
using System.Linq;

class Program
{
        static void Main()
        {
                // Read input from the user
                Console.WriteLine("Enter the array of integers separated by spaces:");
                string input = Console.ReadLine();

                // Convert the input string to an array of integers
                int[] numbers = input.Split(' ').Select(int.Parse).ToArray();

                // Variables to track the longest sequence
                int longestStartIndex = 0;
                int longestLength = 1;

                // Variables to track the current sequence
                int currentStartIndex = 0;
        int currentLength = 1;

        // Loop through the array to find the leftmost longest sequence
        for (int i = 1; i < numbers.Length; i++)
        {
            if (numbers[i] == numbers[i - 1])
            {
                currentLength++;
            }
            else
            {
                if (currentLength > longestLength)
                {
                    longestLength = currentLength;
                    longestStartIndex = currentStartIndex;
                }

                currentStartIndex = i;
                currentLength = 1;
            }
        }

        // Final check in case the longest sequence is at the end of the array
        if (currentLength > longestLength)
        {
            longestLength = currentLength;
            longestStartIndex = currentStartIndex;
        }

        // Print the longest sequence
        Console.WriteLine(“The leftmost longest sequence of equal elements is:”);
        for (int i = 0; i < longestLength; i++)
        {
            Console.Write(numbers[longestStartIndex + i] + “ ”);
        }
    }
}
```

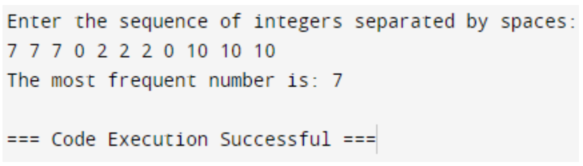


```
Enter the array of integers separated by spaces:
1 2 2 3 3 3 4 4 42 2 3 3 3 4 4 4
The leftmost longest sequence of equal elements is:
3 3 3
=== Code Execution Successful ===
```

7.

```
using System;
using System.Collections.Generic;
using System.Linq;

class Program
{
        static void Main()
        {
                // Read input from the user
                Console.WriteLine("Enter the sequence of integers separated by spaces:");
                string input = Console.ReadLine();

                // Convert the input string to an array of integers
                int[] numbers = input.Split(' ').Select(int.Parse).ToArray();

                // Dictionary to store the frequency of each number
                Dictionary<int, int> frequency = new Dictionary<int, int>();

                // Track the number with the highest frequency
                int mostFrequentNumber = numbers[0];
                int highestFrequency = 0;

                // Traverse through the array to count frequencies
                foreach (int number in numbers)
                {
                        if (frequency.ContainsKey(number))
                        {
                                frequency[number]++;
                        }
                        else
                        {
                                frequency[number] = 1;
                        }

                        // Update the most frequent number
                        if (frequency[number] > highestFrequency)
                        {
                                highestFrequency = frequency[number];
                                mostFrequentNumber = number;
                        }
                        else if (frequency[number] == highestFrequency)
                        {
                                // Check the position of the number to keep the first occurrence
                                if      (Array.IndexOf(numbers,      mostFrequentNumber)      >
Array.IndexOf(numbers, number))
```

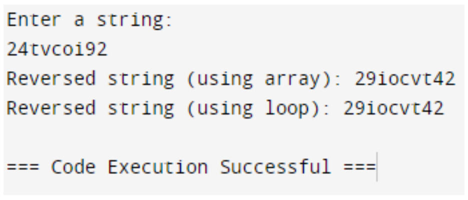
```
                        {
                            mostFrequentNumber = number;
                        }
                    }
                }

                // Print the most frequent number
                Console.WriteLine("The most frequent number is: " +
mostFrequentNumber);
        }
}
```



```
Enter the sequence of integers separated by spaces:
7 7 7 0 2 2 2 0 10 10 10
The most frequent number is: 7

=== Code Execution Successful ===
```

1.

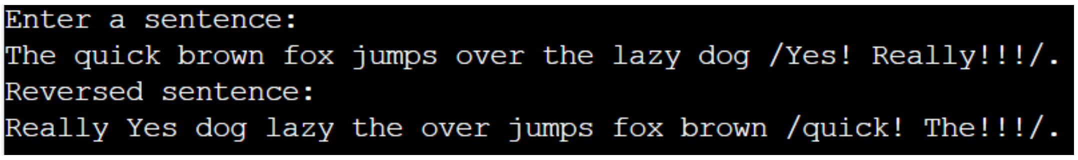
```
using System;

class Program
{
        static void Main()
        {
                // Read input from the user
                Console.WriteLine("Enter a string:");
                string input = Console.ReadLine();

                // Reverse using char array method
                string reversedByArray = ReverseStringUsingArray(input);
                Console.WriteLine("Reversed string (using array): " + reversedByArray);

                // Reverse using for loop method
                Console.Write("Reversed string (using loop): ");
                ReverseStringUsingLoop(input);
        }

        static string ReverseStringUsingArray(string input)
        {
                // Convert the string to a char array
                char[] charArray = input.ToCharArray();
                // Reverse the char array
                Array.Reverse(charArray);
                // Convert the reversed char array back to a string
        return new string(charArray);
    }

    static void ReverseStringUsingLoop(string input)
    {
        // Print the characters of the string in reverse order using a for loop
        for (int i = input.Length - 1; i >= 0; i--)
        {
            Console.Write(input[i]);
        }
        Console.WriteLine(); // For a new line after the reversed string
    }
}
```



```
Enter a string:
24tvcoi92
Reversed string (using array): 29iocvt42
Reversed string (using loop): 29iocvt42

=== Code Execution Successful ===
```

2.

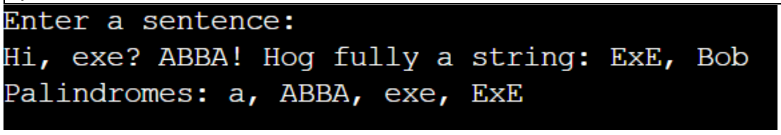
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

class Program
{
    static void Main()
    {
        // Read input from the user
        Console.WriteLine("Enter a sentence:");
        string input = Console.ReadLine();

        // Reverse the ordering of words
        string reversedSentence = ReverseWordsInSentence(input);

        // Print the result
        Console.WriteLine("Reversed sentence:");
        Console.WriteLine(reversedSentence);
    }

    static string ReverseWordsInSentence(string sentence)
    {
        // Define the separators that should not be changed
        string pattern = @“([\s.,:;=()&\[\]”’\\/!?]+)”;

        // Split the sentence into words and separators using regex
        string[] parts = Regex.Split(sentence, pattern);

        // Extract words and separators
        List<string> words = new List<string>();
        List<string> separators = new List<string>();

        foreach (var part in parts)
        {
            if (Regex.IsMatch(part, pattern))
            {
                separators.Add(part);
            }
            else
            {
                words.Add(part);
            }
        }
        words.RemoveAt(words.Count-1);    // the newline character

        // Reverse the order of words
        words.Reverse();

        // Build the reversed sentence
        StringBuilder reversedSentence = new StringBuilder();

        for(int i = 0; i < separators.Count; ++i)
        {
            reversedSentence.Append(words[i]);
            reversedSentence.Append(separators[i]);
        }


        return reversedSentence.ToString();
    }
}
```



```
Enter a sentence:
The quick brown fox jumps over the lazy dog /Yes! Really!!!/.
Reversed sentence:
Really Yes dog lazy the over jumps fox brown /quick! The!!!/.
```

3.

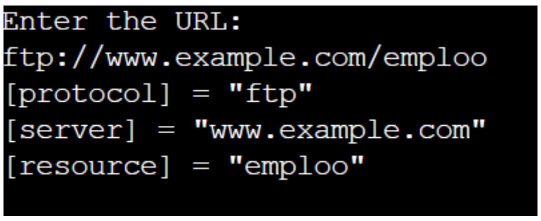
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text.RegularExpressions;

class Program
{
    static void Main()
    {
        // Read input from the user
        Console.WriteLine(“Enter a sentence:”);
        string input = Console.ReadLine();

        // Extract palindromes
        string palindromes = ExtractPalindromes(input);

        // Print the result
        Console.WriteLine(“Palindromes: ” + palindromes);
    }

    static string ExtractPalindromes(string sentence)
    {
        // Define the pattern to match words made of letters
        string pattern = @“[a-zA-Z]+”;

        // Find all matches in the sentence
        MatchCollection matches = Regex.Matches(sentence, pattern);

        // List to store palindromes
        List<string> palindromes = new List<string>();

        // Check each matched word if it is a palindrome
        foreach (Match match in matches)
        {
            string word = match.Value;
            if (IsPalindrome(word))
            {
                palindromes.Add(word);
            }
        }

        // Sort the list of palindromes
        palindromes.Sort();
        // Join palindromes with “,” separator
        return string.Join(“, ”, palindromes);
    }

    static bool IsPalindrome(string word)
    {
        int length = word.Length;
        for (int i = 0; i < length / 2; i++)
        {
            if (word[i] != word[length - i - 1])
            {
                return false;
            }
        }
        return true;
    }
}
```



```
Enter a sentence:
Hi, exe? ABBA! Hog fully a string: ExE, Bob
Palindromes: a, ABBA, exe, ExE
```

4.

```
using System;

class Program
{
    static void Main()
    {
        // Read input from the user
        Console.WriteLine(“Enter the URL:”);
        string url = Console.ReadLine();

        // Parse the URL
        var parsedUrl = ParseUrl(url);

        // Print the result
        Console.WriteLine($“[protocol] = \”{parsedUrl.Protocol}\””);
        Console.WriteLine($“[server] = \”{parsedUrl.Server}\””);
        Console.WriteLine($“[resource] = \”{parsedUrl.Resource}\””);
    }

    static (string Protocol, string Server, string Resource) ParseUrl(string url)
    {
        string protocol = “”;
        string server = “”;
        string resource = “”;

        int protocolEndIndex = url.IndexOf(“://”);

        if (protocolEndIndex != -1)
        {
            protocol = url.Substring(0, protocolEndIndex);
            url = url.Substring(protocolEndIndex + 3);
        }

        int serverEndIndex = url.IndexOf(‘/’);

        if (serverEndIndex != -1)
        {
            server = url.Substring(0, serverEndIndex);
            resource = url.Substring(serverEndIndex + 1);
        }
        else
        {
            server = url;
        }

        return (protocol, server, resource);
    }
}
```



```
Enter the URL:
ftp://www.example.com/emploo
[protocol] = "ftp"
[server] = "www.example.com"
[resource] = "emploo"
```