02 Arrays and Strings

Test your Knowledge

1. When to use String vs. StringBuilder in C# ?

String will create the entire string in the memory while StringBuilder will allocate some buffer space in the memory and applies modifications into that buffer space.

Use a string if only need very few string concatenations. Use StringBuilder for a large number of concatenations.

2. What is the base class for all arrays in C#?

The array class is the base class for all the arrays in C#.

3. How do you sort an array in C#?

Use Array.Sort method.

4. What property of an array object can be used to get the total number of elements in an array?

Array.Length

5. Can you store multiple data types in System.Array?

No.

6. What's the difference between the System.Array.CopyTo() and System.Array.Clone()?

System.Array.Clone() will return a new array containing all the elements in the original array.

System.Array.CopyTo() let you specify an index to the destination array, It copies the elements into another existing array.

Practice Arrays

1. Copying an Array
Write code to create a copy of an array. First, start by creating an initial array. (You can use whatever type of data you want.) Let's start with 10 items. Declare an array variable and assign

it a new array with 10 items in it. Use the things we've discussed to put some values in the array.
Now create a second array variable. Give it a new array with the same length as the first. Instead of using a number for this length, use the Lengthproperty to get the size of the original array.
Use a loop to read values from the original array and place them in the new array. Also print out the contents of both arrays, to be sure everything copied correctly.

```
using System;

0 references
class class1
{
    0 references
    public static void Main()
    {
        int[] arr1 = new int[10] {1,3,5,7,9,11,13,15,17,19};
        int[] arr2 = new int[arr1.Length];
        int i = 0;

        Console.WriteLine("Original Array:");
        for (i = 0; i < arr1.Length; i++)
        {
            arr1[i] = i + 1;
            Console.Write("{0} ", arr1[i]);
        }

        Console.WriteLine();
        Console.WriteLine("The copied Array:");
        for (i = 0; i < arr2.Length; i++)
        {
            arr2[i] = arr1[i];
            Console.Write("{0} ", arr2[i]);
        }


    }
}
```

```
Original Array:
1 2 3 4 5 6 7 8 9 10
The copied Array:
1 2 3 4 5 6 7 8 9 10
```

2. Write a simple program that lets the user manage a list of elements. It can be a grocery list, "to do" list, etc. Refer to Looping Based on a Logical Expression if necessary to see how to implement an infinite loop. Each time through the loop, ask the user to perform an operation,

and then show the current contents of their list. The operations available should be Add, Remove, and Clear. The syntax should be as follows:

```
+ some item
- some item
--
```

Your program should read in the user's input and determine if it begins with a “+” or “-“ or if it is simply “—“ . In the first two cases, your program should add or remove the string given ("some item" in the example). If the user enters just “--“ then the program should clear the current list. Your program can start each iteration through its loop with the following instruction:
Console.WriteLine("Enter command (+ item, - item, or -- to clear)):");

```
class class1
{
    0 references
    public static void Main()
    {
        string command;
        String[] inputs;
        String first;
        String second;
        List<string> animalList = new List<string> { "Cow", "Cat", "Horse", "Camel", "Elephant" };
        Console.WriteLine("Here are the animals that we have:");

        for (int i = 0; i < animalList.Count; i++)
            Console.Write("{0} ", animalList[i]);

        for(int i = 0; i < animalList.Count; i--)
        {
            Console.WriteLine();
            Console.WriteLine("Enter command (+ item, - item, or -- to clear)):");
            command = Console.ReadLine();
            if (command != "--")
            {
                inputs = command.Split(' ');
                first = inputs[0].ToString();
                second = inputs[1].ToString();
                if (first == "+")
                    animalList.Add(second);
                if (first == "-")
                    animalList.Remove(second);
            }

            if (command == "--")
            {
                animalList.Clear();
                Console.WriteLine("We don't have any animals.");
                break;
            }

            Console.WriteLine();
            Console.WriteLine("Now we have the following animals:");
            for(i = 0; i < animalList.Count; i++)
                Console.Write("{0} ", animalList[i]);
```

```
Cow Cat Horse Camel Elephant
Enter command (+ item, - item, or -- to clear)):
+ Tiger

Now we have the following animals:
Cow Cat Horse Camel Elephant Tiger
Enter command (+ item, - item, or -- to clear)):
+ Monkey

Now we have the following animals:
Cow Cat Horse Camel Elephant Tiger Monkey
Enter command (+ item, - item, or -- to clear)):
- Cat

Now we have the following animals:
Cow Horse Camel Elephant Tiger Monkey
Enter command (+ item, - item, or -- to clear)):
- Horse

Now we have the following animals:
Cow Camel Elephant Tiger Monkey
Enter command (+ item, - item, or -- to clear)):
--
We don't have any animals.
```

3. Write a method that calculates all prime numbers in given range and returns them as array of integers

```
static int[] FindPrimesInRange(startNum, endNum)
{
}
```

```
What is the start number?
1
What is the end number?
20
2 3 5 7 11 13 17 19
```

```
class class1
{
    0 references
    public static void Main()
    {
        int counter;
        int i;
        Console.WriteLine("What is the start number?");
        int startNum = int.Parse(Console.ReadLine());
        Console.WriteLine("What is the end number?");
        int endNum = int.Parse(Console.ReadLine());

        for(i = startNum; i <= endNum; i++)
        {
            counter = 0;
            for(int j = 2; j <= i/2; j++)
            {
                if(i%j == 0)
                {
                    counter++;
                    break;
                }
            }
            if (counter == 0 && i != 1)
                Console.Write("{0} ", i);
        }
    }
}
```

4. Write a program to read an array of n integers (space separated on a single line) and an integer k, rotate the array right k times and sum the obtained arrays after each rotation as shown below.
After r rotations the element at position I goes to position (I + r) % n.
The sum[] array can be calculated by two nested loops: for r = 1 ... k; for I = 0 ... n-1.

```
Input          Output          Comments
3 2 4 -1       3 2 5 6         rotated1[] = -1  3  2  4
2                              rotated2[] =  4 -1  3  2
                                    sum[] =  3  2  5  6


1 2 3 4 5      12 10 8 6 9     rotated1[] =  5  1  2  3  4
3                              rotated2[] =  4  5  1  2  3
                               rotated3[] =  3  4  5  1  2
                                    sum[] = 12 10  8  6  9
```

```
using System;

0 references
class class1
{
    0 references
    public static void Main()
    {
        Console.WriteLine("Enter some numbers:");
        string[] inputs = Console.ReadLine().Split();
        int[] numbers = Array.ConvertAll(inputs, int.Parse);
        //Console.WriteLine(String.Join(",", numbers));

        Console.WriteLine("How many rotations you want? Enter the number k:");
        int k = int.Parse(Console.ReadLine());
        int[] sum = new int[numbers.Length];
        for (int i = 0; i < numbers.Length; i++)
            sum[i] = 0;

        for (int r= 1; r < k+1; r++)
        {
            for (int pos = 0; pos < numbers.Length; pos++)
                sum[(pos + r) % numbers.Length] += numbers[pos];
        }

        Console.WriteLine(String.Join(",", sum));
    }
}
```

```
Enter some numbers:
1 2 3 4 5
How many rotations you want? Enter the number k:
3
12,10,8,6,9
```

```
Enter some numbers:
3 2 4 -1
How many rotations you want? Enter the number k:
2
3,2,5,6
```

5. Write a program that finds the longest sequence of equal elements in an array of integers. If several longest sequences exist, print the leftmost one.

Input	Output
2 1 1 2 3 3 2 2 2 1	2 2 2
1 1 1 2 3 1 3 3	1 1 1
4 4 4 4	4 4 4 4
0 1 1 5 2 2 6 3 3	1 1

```
0 references
public static void Main()
{
    Console.WriteLine("Enter some numbers:");
    string[] inputs = Console.ReadLine().Split();
    int[] numbers = Array.ConvertAll(inputs, int.Parse);
    //Console.WriteLine(String.Join(" ", numbers));

    int num = numbers[0];
    int max = 0;
    int counter = 1;
    for (int i = 1; i < numbers.Length; i++)
    {
        if (numbers[i] == numbers[i - 1])
            counter++;
            if (counter > max)
        {
            max = counter;
            num = numbers[i];
        }

        if (numbers[i] != numbers[i - 1])
        {
            counter = 1;
        }
    }
    for (int i = 0; i < max; i++)
        Console.Write("{0} ", num);
}
```

```
Enter some numbers:
1 1 1 2 3 1 3 3
1 1 1
```

```
Enter some numbers:
2 1 1 2 3 3 2 2 2 1
2 2 2
```

```
Enter some numbers:
4 4 4 4
4 4 4 4
```

```
Enter some numbers:
0 1 1 5 2 2 6 3 3
1 1
```

6. Write a program that finds the most frequent number in a given sequence of numbers. In case of multiple numbers with the same maximal frequency, print the leftmost of them

Input	Output
4 1 1 4 2 3 4 4 1 2 4 9 3	The number 4 is the most frequent (occurs 5 times)
7 7 7 0 2 2 2 0 10 10 10	The numbers 2, 7 and 10 have the same maximal frequence (each occurs 3 times). The leftmost of them is 7.

```
public static void Main()
{
    Console.WriteLine("Enter a sequence of numbers:");
    String[] inputs = Console.ReadLine().Split();
    int[] numbers = Array.ConvertAll(inputs, int.Parse);
    //Console.WriteLine(String.Join(" ", numbers));

    int num = 0;
    int counter;
    int max = 0;
    int[] frequency = new int[numbers.Length];

    for (int i = 0; i < numbers.Length; i++)
    {
        counter = 1;
        frequency[i] = -1;
        for (int j = i+1; j < numbers.Length; j++)
        {
            if(numbers[j] == numbers[i])
            {
                counter++;
                frequency[j] = 0;
            }
        }
        if(frequency[i] != 0)
            frequency[i] = counter;
    }

    for (int i = 0; i < numbers.Length; i++)
    {
        if(frequency[i] > max)
        {
            //Console.WriteLine(frequency[i]);
            max = frequency[i];
            num = numbers[i];
        }
    }
        Console.WriteLine($"The number {num} is the most frequent (occurs {max} times)");
    }
```

```
Enter a sequence of numbers:
4 1 1 4 2 3 4 4 1 2 4 9 3
The number 4 is the most frequent (occurs 5 times)
```

Practice Strings

Write a program that reads a string from the console, reverses its letters and prints the result back at the console.
Write in two ways

Convert the string to char array, reverse it, then convert it to string again
Print the letters of the string in back direction (from the last to the first) in a for-loop

Input	Output
sample	elpmas
24tvcoi92	29iocvt42

```
public static void Main()
{
    Console.WriteLine("Enter a string:");
    String inputs = Console.ReadLine();
    char[] charArr = inputs.ToCharArray();
    char[] temp = charArr;
    Array.Reverse(temp);
    Console.WriteLine("After reverse:");
    foreach (char ch in charArr)
        Console.Write(ch);
}
```

```
Enter a string:
sample
After reverse:
elpmas
```

```
Enter a string:
24tvcoi92
After reverse:
29iocvt42
```

2. Write a program that reverses the words in a given sentence without changing the punctuation and spaces
Use the following separators between the words: . , : ; = ( ) & [ ] " ' \ / ! ? (space).
All other characters are considered part of words, e.g. C++, a+b, and a77 are considered valid words.
The sentences always start by word and end by separator.

```
C# is not C++, and PHP is not Delphi!
Delphi not is PHP, and C++ not is C#!


The quick brown fox jumps over the lazy dog /Yes! Really!!!/.
Really Yes dog lazy the over jumps fox brown /quick! The!!!/.
```

```
public static void Main()
{
    Console.WriteLine("Enter a sentence:");
    string inputs = Console.ReadLine();
    String[] words = inputs.Split(',', '.', ' ', '!', '?', '/', ':', ';');
    Array.Reverse(words);
    String results = String.Join(" ", words).ToString();
    Console.WriteLine("After reverse:");
    Console.WriteLine(results);
}
```

```
Enter a sentence:
C# is not C++, and PHP is not Delphi!
After reverse:
 Delphi not is PHP and  C++ not is C#
```

3. Write a program that extracts from a given text all palindromes, e.g. “ABBA”, “lamal”, “exe” and prints them on the console on a single line, separated by comma and space. Print all unique palindromes (no duplicates), sorted

```
Hi,exe? ABBA! Hog fully a string: ExE. Bob
```

```
a, ABBA, exe, ExE
```

```
0 references
public static void Main()
{
    Console.WriteLine("Enter the texts:");
    String[] inputs = Console.ReadLine().Split(',','?','!',':',' ','.');
    char[] temp;
    List<string> results = new List<string>();
    foreach (string word in inputs)
        //Console.WriteLine(words);
        if (word != null)
        {
            temp = word.ToCharArray();
            Array.Reverse(temp);
            string temp1 = new string(temp);
            if (temp1 == word && results.IndexOf(word)==-1)
                results.Add(temp1);
        }
    results.Sort();
    for (int i = 0; i < results.Count; i++)
        if (results[i] != "\t")
            Console.Write("{0}, ", results[i]);
}
```

```
Enter the texts:
Hi,exe? ABBA! Hog fully a string: ExE. Bob
, a, ABBA, exe, ExE,
```

4. Write a program that parses an URL given in the following format:
[protocol]://[server]/[resource]
The parsing extracts its parts: protocol, server and resource.
The [server] part is mandatory.
The [protocol] and [resource] parts are optional.

```
https://www.apple.com/iphone
[protocol] = "https"
[server] = "www.apple.com"
[resource] = "iphone"
```

```
https://google.com
[protocol] = "https"
[server] = "google.com"
[resource] = ""
```

```
ftp://www.example.com/employee
[protocol] = "ftp"
[server] = "www.example.com"
[resource] = "employee"
```

```
www.apple.com
[protocol] = ""
[server] = "www.apple.com"
[resource] = ""
```

```
public static void Main()
{
    String protocol = "";
    String server = "";
    String resource = "";
    Console.WriteLine("Enter an URL:");
    string inputs = Console.ReadLine();
    int indexP = inputs.IndexOf(":");
    if (indexP != -1)
    {
        for (int i = 0; i < indexP; i++)
            //protocol.Insert(protocol.Length, inputs[i].ToString());
            protocol = protocol + inputs[i];
    }

    int indexR = -1;
    for (int i = 2; i<inputs.Length; i++)
    {
        if (inputs[i] == '/' && inputs[i - 1] != '/' && inputs[i - 1] != ':')
            indexR = i;
    }
    if (indexR != -1)
    {
        for (int i = indexR+1; i < inputs.Length; i++)
            resource = resource + inputs[i];
    }
    int startS;
    if (indexP == -1)
        startS = 0;
    else
        startS = indexP + 3;
    int endS;
    if (indexR == -1)
        endS = inputs.Length;
    else
        endS = indexR - 1;
    for (int i = startS; i < endS; i++)
        //server.Insert(server.Length, inputs[i].ToString());
        server = server + inputs[i];

    Console.WriteLine($"[protocol] = {protocol}");
    Console.WriteLine($"[server] = {server}");
    Console.WriteLine($"[resource] = {resource}");
}
```

```
Enter an URL:
ftp://www.example.com/employee
[protocol] = ftp
[server] = www.example.com
[resource] = employee
```

```
Enter an URL:
https://google.com
[protocol] = https
[server] = google.com
[resource] =
```

```
Enter an URL:
www.apple.com
[protocol] =
[server] = www.apple.com
[resource] =
```