02 Arrays and Strings

Test your Knowledge

1. When to use String vs. StringBuilder in C# ?
2. What is the base class for all arrays in C#?
3. How do you sort an array in C#?
4. What property of an array object can be used to get the total number of elements in an array?
5. Can you store multiple data types in System.Array?
6. What's the difference between the System.Array.CopyTo() and System.Array.Clone()?

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 *Length*property 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.

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)):");
```

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

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

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
```

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
```

7. 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.
```

Practice Strings

1. 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
```

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!!!/.
```

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
```

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"

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

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

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

Explore the following Topics

- Strings
- Arrays
- Using the StringBuilder