# Operations on Arrays

https://csci-1301.github.io/about#authors

June 21, 2021 (08:04:48 PM)

## Contents



# 1 Operations on numeric arrays

Start by creating a new C# solution.

After creating the solution, declare and initialize an int array called `numbers`. Initialize the array so that it holds the following values, in the same order:

```
4, 2, 6, 1, 7, 5, 3, 4, 2, 2, 8, 6, 3, 11, 7, 2, 9, 3, 1, 9, 7
```

## 1.1 Displaying values

After declaring and initializing `numbers` array, write statements to:

1. Display every value left to right
2. Display every value at even indices (skip odd indices)
3. Display all values that are greater than 5

## 1.2 Counting values

Next, write statements that provide answers to following questions:

1. Count the sum of all `numbers` then display the result. (The expected value is 102)
2. Count how many times value 7 occurs in `numbers`, then display that value. Check that your program outputs the correct answer, which you can determine by visually observing the array values.

### 1.3 Finding values

Now implement the following statements:

1. Find the *index* of first 7, then display that index. If the value does not exist, display -1 to indicate it was not found. Check that your solution is correct by comparing what you obtain from the program with what you know by visually observing the array.
2. Find the maximum value in `numbers`. Check that the solution you implement obtains the expected value.

### 1.4 Evaluate your solution

After implementing these methods and assuming the program you implemented obtained the expected values, *ideally* the solution should still work even if the values stored in `numbers` array change, or even if the array length changes.

To test your solution, go back to the beginning of the program where you declared `numbers` array, then change the initialization so that the new array values are:

```
55, 92, 12, 90, 37, 18, 6, 20, 80, 18, 46, 19, 65, 68, 18
```

Then re-run the program.

Check that you obtain expected values:

- the sum should now be 644
- since 7 does not occur in the array anymore,
  - count should be 0
  - first index of 7 should be -1
- maximum value is now 92

## 2 Working with two arrays

For this part of lab, create two `char` arrays, with following values:

```
char[] chars1 = {'K', 's', 'Q', 'U', 'i', 'N', 'K', 'N', 'h', 't', 'u'};
char[] chars2 = {'?', 'E', 'U', 'a', 'j', 'X', 'L', 'G', '@', 'L', 'l', 'C', 'w', 'J',
 ↪  'U' };
```

Next, write statements that answer the these two questions:

1. Does value `w` occur in both arrays (true/false)?
2. What is the first value that occurs in both arrays (searching left to right)?

After completing these two problems, make sure the program answers these questions correctly. The expected results are:

- Does `w` occur in both arrays -> **`false`**
- first value that occurs in both arrays -> `U`

Again, evaluate your work by changing the array initialization to:

```
char[] chars1 = {'s', 'p', 'd', 'P', 'y', 'D', 'w', '?'};
char[] chars2 = {'V', 'D', 'l', 'P', 'w', 'O', 'y', 'k', 'D', 'Z' };
```

Then run the program again.

Ideally the program should not crash and should still produce correct results:

- Does `w` occur in both arrays -> `true`
- first value that occurs in both arrays -> `P`

If the program does not produce these expected answers after changing the array values, review your program and try to determine how to write a solution that works for *any* array values.