Array Practice

Exercise 1 – Max and Min

Write a program that will find the maximum and minimum values in an array. You can "hardcode" the array at the beginning for now, but your application should work for any array, of any size, by just changing the list:

```
int values = {1, 2, 6, 3, 7, -9, 5, 7, 4, 3, 11, 25, 7, 6};
```

Exercise 2 – Insert a character

Write a program that will accept a **String str** of size n, a **char ch**, and a position **int x** (**x** between 0 and n).

Your program will convert **str** to a **char[]** called **original** and create a new **char[]** called **modified** of size n+1. It will copy all the **char**s from **original** into **modified** except that **ch** will be in position **x** in **modified**, and the remaining **char**s will be shuffled along to fill in **modified**. Output **modified** as a **String**.

For example, suppose

str = "gr!"

ch = '8'

x = 2

Then n=3, n+1=4, **original = {'g','r','!'}**.

Your program will generate **modified={'g','r','8','!'}** and output the **String "gr8!"**.

Exercise 3 – Counting Digits in an Array

Write a program that takes an array of 1-digit ints (like **{0,5,1,6,5,5,2,3,9,7,1,0}**) and counts the number of occurrences of each digit. The array should be hard-coded in the program.

Exercise 4 – Counting Digits in a Number

Write a program that takes a **long** as input and counts the number of occurrences of each digit.

Use an array of **int** of size 10 to store the number of occurrences. Increment the correct value in the array for each digit.

At the end, output the count for each digit. For example, if the input was **458660656** you would output something like this:

```
0: 2
1: 0
2: 0
3: 0
4: 1
5: 2
6: 3
7: 0
8: 1
9: 0
```

Exercise 5 – Transposing Pairs of Chars

Write a program that takes a **String** as input and outputs a new **String** with every pair of **char**s reversed. For example, if the input was

```
Pseudoscience
```

the output would be

```
sPueodcseicne
```

If there are an odd number of letters, don't change the last one.

Build your output as a **char[]** and then convert it to a **String**.

Note that if you put the modified **String** back into your program as input, it should produce the original **String** as output.