

CS 116

Programming Assignment 4

This assignment is to make sure that you understand the basics behind the chapters covered so far. The important thing, before you start this assignment, is to understand conditionals, iterations, functions, Arrays and Strings really well. The estimated time for completion is about five hours.

In this assignment, you are given a Driver function that tests four other functions (**Exercise1**, **Exercise2**). You have to write both functions in a way to pass the tests. Every test has a specific output; if you get it then you passed the test.

Copyright Notice

Copyright © 2010 DigiPen (USA) Corp. and its owners. All rights reserved.

No parts of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language without the express written permission of DigiPen (USA) Corp., 9931 Willows Road NE, Redmond, WA 98052

Trademarks

DigiPen® is a registered trademark of DigiPen (USA) Corp.

All other product names mentioned in this booklet are trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Details

Code Details:

In this assignment, we don't use global variables and **of course** you should not add any. We have a main function called **"Driver"** that calls functions in order to test them. Your job is to create those functions so that the tests pass.

Functions to implement:

Function Name	Parameters	Return Type
Exercise1	aNumbers: Array that contains all the numbers. iNumberOfMinsToRemove: An integer that specifies how many minimums we want to remove from the aNumbers array	nothing

Function(Exercise1) Description

This function's role is to remove 0 or more minimum values from an array of numbers. The user will send a array filled with numbers (**aNumbers**) and will specify how many minimum numbers (**iNumberOfMinsToRemove**) he wants to remove.

Ex: if **aNumbers** contains 1,0,6,0,4 and **iNumberOfMinsToRemove** is 1 the result should be 1,6,0,4

The user can specify positive or negative **iNumberOfMinsToRemove** values. If he specifies a negative value, then you should start removing numbers from the end.

Ex: if **aNumbers** contains 1,0,6,0,4 and **iNumberOfMinsToRemove** is -1 the result should be 1,0,6,4

Check the tests in the Driver function to get a better understanding.

Function Name	Parameters	Return Type
Exercise2	<p>aData: Array of strings containing all the data (name, last name and age in them)</p> <p>uiSortingType: An unsigned integer that specifies the type of sorting the user wants.</p> <p>bDescending: Boolean that specifies if the user wants to sort in a Descending or Ascending way.</p>	nothing

Function(Exercise2) Description

This function's role is to sort an array filled with strings (**aData**) according to the user's specified sorting types.

Each string inside the array slot contains:

A **name**, a **last name** and an **age**

The user can specify, using the **uiSortingType** variable, the way he/she wants to sort the array.

Sending **0** means sorting according to the **name**.

Sending **1** means sorting according to the **last name**.

Sending **2** means sorting according to the **age**.

Sending **Anything Other Than 0,1 or 2** means using the normal string sorting.

One more thing that the user can specify, using the **bDescending** Boolean variable, is whether he/she wants the sorting to be ascending or descending. **True** means descending and **false** means ascending.

Check the tests in the Driver function to get a better understanding.

Special Notes:

- You will need to use the customizable sort function for this exercise, so you will need to create your own sorting function rules.
- In order to change a String to an Integer you should typecast it:

```
var sTest:String = "13";
var iTestAsInt:int = int(sTest);
```

- Assume we are sorting by age, the following two outputs are considered the same:
Charlotte Smith 13,Imad Hanna 26,Chris Doe 26,Charlotte Smith 59
Charlotte Smith 13,Chris Doe 26,Imad Hanna 26,Charlotte Smith 59

Comments

In this and future assignments, you are required to include:

- A file header comment at the beginning of the fla. The format is shown in the "Comments.flc" file given to you in the beginning of the semester and should be present at the very top of all your code.
- Function header for each function you create. The format is shown in the "Comments.flc" file given to you in the beginning of the semester and should be present at the top of every function.
- Inline commenting for your code.

What to submit

You must submit the fla file (**Arrays.flc**) in a single .zip file named correctly (go to the class page on moodle and you will find the assignment submit link). **Do not submit any other files than the ones listed.**

If you've forgotten how to submit files, the details about how to submit are posted in the syllabus. Failure to follow the instructions will result in a poor score on the assignment (and possibly a zero).

Special note:

The due date/time posted is the positively latest you are allowed to submit your code. Since the assignments can easily be completed well before the deadline, you should strive to turn it in as early as possible. If you wait until the deadline, and you encounter unforeseen circumstances (like being sick, or your car breaking down, or something else), you may not have any way to submit the assignment on time. Moral: **Don't wait until the last day to do your homework.**