Assessment Sheet 5

This is the fifth of eight assessment sheets that you will be given over the duration of this unit. You will need to complete the tasks as outlined below and then document them in a word document. At minimum, you should provide screen shots of:

- Your code
- The output that your code generates

In instances where your code could give different outputs depending on what values it is given; you should provide multiple screen shots of the console screen showing different outputs to demonstrate that the code is working correctly.

This assessment sheet will focus around all the concepts you have learnt up until now, especially arrays

To achieve a D grade

Successful completion of this task (documented in your portfolio) will allow you to achieve up to a D1 grade.
Please follow all instructions:

For this task, we will look at creating a mini mathematical program that utilises arrays. This is going to be similar to the menu program you created for a previous assessment sheet.

Please follow the steps below:

- Create an array and initialise it with the following numbers:
  8, 6, 9, 15, 22

- As with assessment sheet 3, you are going to display a menu.
  Make the program print out the following:

  "Please enter your choice from the following menu:
  A – Repopulate array
  B - Display all values
  C – Replace one number
  D – Calculate the mean"

- Get the user to input their menu choice via the keyboard.

- If the user enters A then the system should allow them to enter new numbers (via the keyboard) that will be stored in the array. All numbers in the array should be replaced, this will involve a loop.

Author: Darren Cunningham
Version 1.0

- If the user enters B then the contents of the array should be displayed on the screen.

- If the user enters C then the system should ask the user what element of the array they want to replace. It should then ask what number they want to put in their chosen array element.
  The program should then replace the users chosen array element with the value they entered.

- If the user enters D then the mean of all the numbers in the array should be calculated and the answer should be displayed on the screen

Complete the above task and then document it in your portfolio.

Note, to achieve a D

- The write up must demonstrate that the code works
- For a D grade, you only need to provide screen shots of your code and some of the different outcomes it can give - At this stage you do not need to do any writing
- Your word document should have appropriate headings to ensure that this task can easily be identified alongside your work from other assessment sheets.

To achieve a C grade

Modify the above program so it now behaves as follows:

- Add the following to the menu:

  "E – Find largest number
  X - Exit"

- Modify the program so that if the user enters E then the program should find the largest number stored in the array and then display it on the screen

- The whole program should be put in a loop so that once the user has performed an action, the program will display the menu again and allow the user to select another option. The program should only end if the user enters X as their menu choice

- If the user enters an invalid menu choice, then they should be informed of their mistake

- Both uppercase and lowercase letters should be accepted for menu choices.

- As always, with chars, any erratic behaviour should be fixed.

Add the modifications to your portfolio

Author: Darren Cunningham
Version 1.0

Furthermore, to achieve a C

- The above task should be completed and documented in the portfolio. The portfolio should demonstrate that the code works
- ALL tasks must be accompanied by written descriptions or annotations. These must show some basic understanding of what the code is doing

To achieve a B grade

Modify the above program (C grade section) so it now behaves as follows:

- Change the array so that it is now of size 100 and does not contain any values.

- Modify the program so that before the menu is displayed, the user is forced to enter values in to the array.
  The user should first be asked how many values they want to enter.
  Then the program should get these values from the user and add them to the array

  For example, if the user enters that they have 5 values, the system will then ask them 5 times to enter a number to be placed in the array

  Once the user has finished entering their values, the system should add a -1 in to the next free element of the array. This will act as a 'terminator'

- The program should then run as before:
  The menu is displayed, the user enters what they want to do, the chosen action is performed and then the system loops back around to display the menu again.
  Be careful about what you place inside the main loop. The system should not force the user to enter values each time the program loops.

- Modify all menu actions so that they now work with the terminator.

- The repopulate array functionality (if the user enters A as their menu choice) needs to be changed so that it now works as described in the second bullet point of this section. I.e. the user should be asked how many values they wish to enter. These values should be retrieved from the user and then a -1 should be placed in the next free location of the array.

To achieve a B:

- The above task should be completed and documented in the portfolio. The portfolio should demonstrate that the code works
- All tasks must be accompanied by written **explanations** - All new concepts, as they are encountered should be explained.
  This is in addition to the C grade tasks where you describe what your code is doing
- Any concepts that have been explained in work covered by previous task sheets do **not** need to be explained again (i.e. variables)

To achieve an A grade

- Your written explanations from the B task should be very detailed

As before, the A grade task will require some independent research:

- Demonstrate any alternative approaches you may have found and explain them
- For the menu choice, you should receive the users input using a function called **getchar** rather than scanf. Look in to getchar, demonstrate it and explain it.
- Please give this section an appropriate heading to ensure it is easy to find.

Author: Darren Cunningham
Version 1.0