SIT102 Introduction to Programming

Pass Task 6.1: Working with Arrays and Vectors

Overview

This task comprises two parts. In part A, you will create a simple program which will create and work with an array of strings. This array will then be populated with values, printed out to the console, and then, we'll use the values to calculate some basic statistics. In part B, you will change your working with arrays program (part A) by using a dynamic array (vector in C++) to keep track of a number of values for the user.

Submission Details

Use the instructions on the following pages to create a program to explore the use of arrays and vectors.

Submit the following files to OnTrack.

- Your program code
- Screenshots of your program outputs from both parts A and B
- Answers to the associated questions

The focus of this task is on the declaration and use of standard static and dynamic arrays (using the vector class in C++).

Instructions

Let's get started.

Part A 1. Create your own *Working With Arrays* project, and use your custom terminal user input code file and header (or copy in the code of your user input functions if you have not done these credit tasks).

1. Implement the features from the weekly videos including:

    - Read and display a list of names.

    - Create a function to calculate the **total length** of the list of names.

    - Create a function to return the **longest name**.

    - Create a function to return the **index of** a given name within the list. return the index of a given name in the array. This should return -1 if the name is not present.

      `int index_of(string value, string data[], int size) {...}`

    - Include a `void print_summary(string data[], int size)` procedure to do the following:

        - Print all names
        - Print total length of all names
        - Print the longest name
        - Print the index of your name

    - Have `main` create and work with an array of strings

        - Use a constant for the size (make it 3)
        - Loop to read in the names
        - Call `print summary` to output the details
        - Allow the user to **change** one of the names - ask them to enter the name to change, and then the new name. If the name to change was in the list, then replace the value at that index with the new name.
        - Call `print summary` again to output the new details

2. Run your program twice to test both with and without your name in the array, then grab a screenshot of the output and paste it on a document file. Later on, your document will store one more screenshot from part B.

Part B 1. Watch the topic 6 videos on dynamic arrays and working with vectors.

2. Return to your own *Working With Arrays* project (part A), and make the same changes as shown in the videos:

    - Include the `vector` library
    - Make sure you have `using namespace std;`
    - Change all of the array parameters to make use of `vector<string>`

    - Remember to use `.size()` to get the size of the vector.
    - Ask the user how many names they want to add, and loop that many times adding one name each loop.
    - Use `.push_back(...)` to add data to the end of the vector.
    - Remember to use pass-by-reference with all of the vector parameters: eg. `void print_summary(const vector<string> &data) { ... }`.

3. Compile and run to make sure your program works, then grab a screenshot of the output and paste it into your document for storing all screenshots. Your document will be submitted to OnTrack in pdf format.

4. Download the resources associated with this task and answer the questions. You should now be ready to submit.

Task Discussion

Discuss the following with your tutor to demonstrate your understanding of the concepts covered.

- Explain how arrays work - how do they enable you to easily work with a list of values in code?
- Explain the role of the `for` loop in working with arrays
- Discuss how the index of an array works. For example: Do you always need to loop through the array? Could you go backwards through an array (from end to start)? Explain how you can use an index if you want to know where one particular value is.
- Explain the similarities and differences between static arrays and vectors.