CPE 150 Laboratory 12: Pointers and Strings I

Department of Computer Engineering
Yarmouk University

Summer 2017

1 Objectives

- To use pointers. Pointers enable programmers to simulate passing arguments call-by-reference.
- To understand the close relationships among pointers, arrays and strings. The variable name of an array or a string is a pointer to the address at which that array or string is stored.
- To declare the use of arrays of strings. Strings are used to store and manipulaye character data. In C++, strings are stored as arrays of charcters. The string manipulation library, `cstring`, is used to compare, copy, tokenize, and determine the length of strings.

2 Lab Exercise 1 - Median Function

In statistics, when a set of values is sorted in ascending or descending order, its median is the middle value. If the set contains an even number of values, the median is the mean, or average, of the two middle values. Write a function that accepts as arguments the following:

1. An array of integers
2. An integer that indicates the number of elements in the array

The function should determine the median of the array. This value should be returned as a double. (Assume the values in the array are already sorted.)

Note: Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.

3 Lab Exercise 2 - Reverse Array

Write a function that accepts an `int` array and the array's size as arguments. The function should create a copy of the array, except that the element values should be reversed in the copy. The function should return a pointer to the new array. Demonstrate the function in a complete program.

Note: Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.

4 Lab Exercise 3 - Magic Function

Implement a function `void fill_with_values(int* a, int size, int* f)` that sets the i-th element of the array to $f(i)$. Here `int* f` is a function pointer to a function that consumes an `int` and yields an `int`. Provide a `main` function in which you call the `fill_with_values` function so that an array of ten integers is filled with $1, 4, 9, 16, 25, \cdots, 100$.

Note: Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.

5 Postlab Exercise

Write a program that uses a menu-driven interface. The program should offer the user five options as follows (these should be displayed on the screen):

```
Enter a choice:
0 Enter grades
1 Print the array of grades
2 Find the minimum grade
3 Find the maximum grade
4 Print the average on all tests for each student
5 End program
```

One restriction on using arrays of pointers to functions is that all the pointers must have the same type. The pointers must be to functions of the same return type that receive arguments of the same type. Store the pointers to the four functions in array `processGrades`, and use the choice made by the user as the subscript into the array for calling each function. Option 0 will allow the user to enter the grades in already predefined array of pointers, i.e., two-dimensional array. Each row corresponds to a student and the column corresponds to the a grade for a specific course. You can assume that we have 20 students and 5 courses per student. Option 1 prints the array of grades. Options 2 and 3 finds the minimum and maximum grades. Option 4 computes the avergae grade for each student. Finally, option 5 quits the program.

Note: Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.