CSC10002 – Programming Techniques

Lab 03
Dynamic Array

Cảm ơn thầy Trần Duy Quang đã cung cấp template cho môn học

Department of Software Engineering-FIT-VNU-HCMUS

1

Notes

Create a single solution/folder to store your source code in a week.

Then, create a project/sub-folder to store your source code of each assignment.

The source code in an assignment should have at least 3 files:

- A header file (.h): struct definition, function prototypes/definition.
- A source file (.cpp): function implementation.
- Another source file (.cpp): named YourID_Ex01.cpp, main function. Replace 01 by id of an assignment.

Make sure your source code was built correctly. Use many test cases to check your code before submitting to Moodle.

Name of your submission, for example: **20125001_W03_04.zip**

2

Content

In this lab, we will review the following topics:

- How to use a pointer as a 1D array.
- How to use a pointer as a 2D array.

3

Assignments

A: 2 problems / assignments.

H: 6 problems / assignments.

3.1 Assignment 1

In the C programming language there is no pass-by-reference syntax to pass a variable by reference to a function. Instead a variable is passed by pointer (just to be confusing, sometimes passing by pointer is referred to as pass by reference). This Practice Program asks you to do the same thing as C, which in practice would be simpler to implement using C++'s reference parameter syntax. Here is the header for a function that takes as input a pointer to an integer:

```
void addOne(int *ptrNum);
```

Complete the function so it adds one to the integer referenced by ptrNum. Write a main function where an integer variable is defined, give it an initial value, call addOne, and output the variable. It should be incremented by 1.

3.2 Assignment 2

The following function uses reference variables as parameters. Rewrite the function so it uses pointers instead of reference variables, and then demonstrate the function in a complete program.

```
int doSomething(int &x, int &y) {
        int temp = x;
        x = y * 10;
        y = temp * 10;
        return x + y;
}
```

3.3 Assignment 3

This Practice Program requires that you read the optional section about pointer arithmetic. Complete the function isPalindrome so that it returns true if the string cstr is a palindrome (the same backwards as forwards) and false if it is not. The function uses the cstring library.

```
bool isPalindrome(char* cstr)
{
   char* front = cstr;
   char* back = cstr + strlen(cstr)-1;
   while (front < back)
   {
      // Complete code here
   }
   return true;
}
```

3.4 Assignment 4

In statistics, the mode of a set of values is the value that occurs most often or with the greatest frequency. Write a function that accepts as arguments the following:

A) An array of integers

B) An integer that indicates the number of elements in the array

The function should determine the mode of the array. That is, it should determine which value in the array occurs most often. The mode is the value the function should return. If the array has no mode (none of the values occur more than once), the function should return −1. (Assume the array will always contain nonnegative values, a[i] is from 0 to 1.000.000)

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

3.5 Assignment 5

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:

A) An array of integers

B) 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.)

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

3.6 Assignment 6

Write a function that accepts an int array and the array's size as arguments. The function should create a new array that is twice the size of the argument array. The function should copy the contents of the argument array to the new array and initialize the unused elements of the second array with 0. The function should return a pointer to the new array.