

[CS302-Data Structures] Homework 1: C++ Review

Instructor: Kostas Alexis

Teaching Assistants: Shehryar Khattak, Mustafa Solmaz, Bishal Sainju

Fall 2018 Semester

Exercise 1. Given the function definition

```
void twist(int a, int& b)
{
    int c;

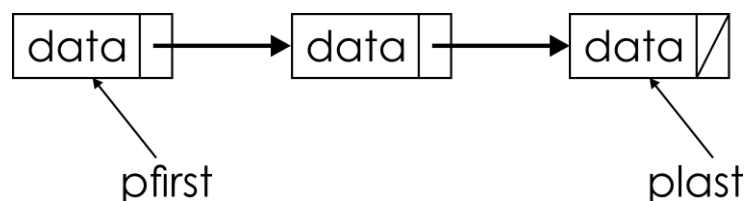
    c = a + 3;
    a = a * 3 + c;
    b = c + a;
}
```

what is the output of the following code fragment that invokes `twist`? (assuming that all variables are of type `int`). Explain your answer.

```
r = 2;
s = 3;
t = 3;
twist(t, s);
cout << r << ' ' << s << ' ' << t << endl;
```

Exercise 2. (a) When passing a static 2D array to a function, the number of columns must be specified in the function prototype and function heading. Explain why. (b) Give the C++ statements for the dynamic allocation of an array A with 3 rows and 5 columns. Draw a diagram that illustrates the structure of the dynamic array in memory. Carefully explain how the system would find the address of statement `A[i][j]` using the pointer structure associated with the array.

Exercise 3. Using C++ object oriented programming, create a singly linked list. The singly linked list is a chain of its item. Each item contains two parts—data and pointer. The data part is to store information and the pointer part of an item is used to point or to store the address of the next item.



You are asked to build a singly linked list that has two pointers—one(*pfirst*) points to the first item of the list and another one(*plast*) points to the last item of the list. The program should also provide a menu of choices that the user can use to do basic operations on the linked list such as a) adding a new item, b) delete an item, c) show the number of items, d) show all items, or simply to exit.

Exercise 4. The following program has careless errors on several lines. Find and correct the errors and show the output where requested.

```
#include <iostream>

int main()
{
    int* ptr;
    int* temp;
    int x;

    ptr = new int;
    *ptr = 4;
    *temp = *ptr;
    cout << ptr << temp;
    x = 9;
    *temp = x;
    cout << *ptr << *temp;
    ptr = new int;
    ptr = 5;
    cout << *ptr << *temp; // output is: ...
    return 0;
}
```