

ESE 224 Homework 4

Due: Thursday, Nov. 10 at 11:59pm (2 weeks)

*To receive full credit you must submit your code (.cpp and .h files) **AND** screenshots of your code with outputs from the console of that code in a separate document.* (5 points for screenshot report)

Dynamic Memory Allocation

1. (20 points) Dynamically allocate a character array and input a string from the keyboard. Replace all 'g' in the string with 'b'. Replace all 'b' with 'a'.

Ex: "globe" should turn into "bloae"

No other extra characters are supposed to be printed when displaying the output

Press enter key for the next string

You are using dynamic memory allocation, (new) don't forget to deallocate it with delete.

Linked List

2. (25 points) A car-recycling company keeps track of its cars by giving each car a label (an integer) based on the type. If there is more than one car of the same type, **ALL** the cars of the same type are recycled (destroyed to give scrap metal). **All the cars are already labelled and placed in ascending order.** Find the cars that are to be spared (unique cars).

Create a **singly linked list** with data that represents the labels of the cars. So, you must insert nodes in a linked list in an ascending order with some duplicates to represent cars of the same type.

Remove any nodes with duplicate numbers from the sorted linked list, when the head of the linked list is given. This means you must search the list for duplicates and delete them. Leave only the unique numbers from the initial list. Print/Return the contents linked list after the deletion of duplicate elements.

Example:

Elements of the list: 1 1 1 2 3 3 3 4 5 6 6 6 6 7 8 8 8

Elements of the list after deletion: 2 4 5 7

The number of nodes in the list is in the range [0, 100].

Remember, if you are using dynamic memory allocation, (new) don't forget to deallocate it with delete.

Queue Data Structure

3. (25 points) Create a class queue and its member functions. Write a program to execute all the functions of a queue and create a menu type for the user giving the user options. Print the queue after each and every command of the user's input. Below given is an example:

```
Queue Main Menu
1.Insert
2.Remove
3.Display
Others to exit
Enter Your Choice: 1
Enter The Value to be Insert: 1
```

Stack Data Structure

4. (25 points) The Tower of Hanoi is a mathematical puzzle. It consists of three poles and a number of disks of different sizes which can slide onto any poles. The puzzle starts with the disk in a neat stack in ascending order of size in one pole, the smallest at the top thus making a conical shape. The objective of the puzzle is to move all the disks from one pole (say 'source pole') to another pole (say 'destination pole') with the help of the third pole (say auxiliary pole).

Link to an example game can be found here: <https://www.mathsisfun.com/games/towerofhanoi.html>

The puzzle has the following two rules:

1. You can't place a larger disk onto a smaller disk
2. Only one disk can be moved at a time

HINT:

The algorithm for the Tower of Hanoi problem is the following:

1. Calculate the total number of moves required i.e. " $2^n - 1$ " (n is the number of disks.)
2. If the number of disks (i.e. n) is even then interchange the destination pole and auxiliary pole.
3. for $i = 1$ to total number of moves:
if $i \% 3 == 1$:
legal movement of top disk between source pole and destination pole
if $i \% 3 == 2$:
legal movement top disk between source pole and auxiliary pole
if $i \% 3 == 0$:
legal movement top disk between auxiliary pole and destination pole