# CS 400 HW 4: Linked List and Stack

**Note**: for CS 400’s homework, you will need to

Submit your solution(.cpp file) to the blackboard

## Question 1:

Design a function that takes in a string of (mixed) parentheses and returns whether the given string is balanced.

Note that you must deal with two parenthesis types: round ones () and angle brackets <>.

---

For example:

• <>()<()>(<>) is balanced. Note that you may use () to include <>, or vice versa.

• (<)> is not: you cannot cross-use different types.

• ><>< is not: you cannot have closing > before opening <.

• ()) is not: too may closing ).

• <<> is not: too many openings <.

• <><<>>()(()) is balanced.

## Question 2:

Create a function for converting a given decimal number, print out the hexadecimal representation.

Note that this is similar to the function for converting a decimal to binary (we covered during the class).

## Question3: implement a paylist by using LikedList

Consider a scenario where you need to implement a playlist for a music player. Each song in the playlist needs to be stored in a specific order, and you should be able to add new songs, remove sons, and traverse through the playlist.

In this case, a linked list can be a suitable data structure for implementing the playlist. Each node in the liked list represents a song, and the “next” pointer of each node points to the next song in the playlist.

// TODO: according to the PlayList class created,

// TODO: implement all the member functions in the same file.

Use the code from main() to test your result.

* work on the CS400LinkedListApplication(PlayList)\_name\_WSUID, and
* Submit your solution to the blackboard.