EE 312
Assignment 6
Due date: April 8th 2016

In this assignment you will implement several recursive functions. The aim of the assignment is to help you practice and master recursion.

You are given two files as part of this assignment. The main.c file will only be used to test your code. You are responsible to test your own code. No test cases are given as part of this assignment. The main.c file will **not be graded**. Do not implement your functions in main.c

The assignment-7.c file includes functions that you need to implement.

Deliverables

As usual, you will submit your code through Assembla. You need to commit both the main.c and the assignment-7.c files.
Make sure to submit the files before the deadline.

Part I: Generate Strings of Matched Parens

In this part of the assignment, you will write a program that takes a given number *n* and generates all string with matched parens that contains *n* left parens "(" and *n* right parens ")".
For example, "()", "(())" and "()()" are strings with matched parens. ")(" and "(()" are not string with matched parens.
Your function takes *n* as a parameter and generates all string with matched parens having *n* left parens and *n* right parens.
For example, for n = 2 your program should generate: "(())" and "()()".
Your program will need to print the strings with matched parens to STDOUT (console).
Print each string with matched parens on a new line. The output for the above example should be:
(())
()()
The order in which you print the strings does not matter.

Part II: Generate All Palindromic Decompositions

A palindrome is a string that reads the same forwards and backwards. For example, "racecar" is a palindrome.
In this part of the assignment, you will generate all palindrome decomposition of a given string. The decomposition of a string is a set of strings whose concatenation is the string itself. For example, "a bc" is one decomposition of the string "abc".
In this part of the assignment, you are given a string and you need to generate all the decompositions of that string such as each part of the decomposition is a palindrome. For example, given the string "abadefe" one palindromic decomposition is "aba d efe" since each of the decomposition "aba", "d" and "efe" are palindromic strings. There might be more than one palindromic decomposition, and your program should print all palindromic decomposition.
When printing one decomposition, separate the different components by a SPACE.
Print each single decomposition on a new line.