CE 242 - Data Structures and Algorithms
Spring 2009 - Ahmet Ardal
Lab Assignment 3

1. Using the reference stack data structure implementation, write code for the implementation of StackPush() and StackPop() functions. The code for those functions' declarations and definitions are provided in the Stack.h and Stack.cpp source files and what you are asked to do is just filling in the function definitions. Further explanation is given as inline code comment in the Stack.h and Stack.cpp source files.

2. Write a function that takes a string as input and determines whether or not the parentheses, braces and square brackets in the input string are well-formed. That is; every opening brace, square bracket or parenthesis should have a corresponding closing brace, square bracket or parenthesis, respectively. The function should return true if the parentheses are well-formed, otherwise false. In order to check the correctness of the function you will implement, provide a test code that calls your function with a sample string containing braces, square brackets and parentheses. For instance; the function should return true for the string "{(abc[x]-y{z})}" and should return false for the string "{(abc[x]-[y{z}))".

(Hint: You should make use of the Stack data structure in order to solve the problem.)

Function prototype:

```
bool isParenthesesWellFormed(const char *str);
```

3. Write a function that takes a string as input and prints the words in the string in reverse order. Note that the words in the string are expected to be separated by space characters. In order to check the correctness of the function you will implement, provide a test code that calls your function with a sample string containing words separated by space characters. For instance, the function should print the string "print me now" as "now me print".

(Hint: You should make use of the Stack data structure in order to solve the problem.)
(Hint: You can replace space characters with null characters in order to separate words of the string.)

Function prototype:

```
void printWordsReverse(char *str);
```