Lab # 6

Task 1:

Write a program that will prompt the user to enter a string of decimal digits, ending with a carriage return and print the sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again.

Sample execution:

Enter a decimal digits string: **1299843**
The sum of the digits in hex is: 0024

Note: Use Stack for Task 2 and Task 3

Task 2:

Write an Assembly Language Program that lets the user type some text, consisting of words separated by blanks, ending with a carriage return, and displays the text in the same word ordered as entered, but with the letters in each word reversed.

Sample execution:

Enter a string: **COAL is an interesting Subject**
You Enter: **LAOC si na gnitseretni tcejbuS**

Task 3:

Write an Assembly Language Program that lets the user type in an algebraic expression, ending with a carriage return that contains parenthesis only. As the expression is being typed in, the program evaluates each character. If at any point the expression is incorrectly bracketed [too many left or right], the program tells the user to start over. After the carriage return is typed, if the expression is correct, the program displays "expression is correct" and the program asks the user if he or she wants to continue. If the user types 'Y', the program runs again. If the expression is not correct, the program displays "too many left brackets, begin again" or “too many right brackets, begin again” according to expression. Your program does not need to store the input string, only check it for correctness.

Sample execution:

Enter an algebraic expression: **a+b)**
Too many right brackets, Begin again
Enter an algebraic expression: **a+(b-c**
Too many left brackets, Begin again
Enter an algebraic expression: **a+(b-c)**
“Expression is correct”
Type Y if you want to continue: **N**