**All homework assignments turned in for credits are to be the result of individual effort.**

**Please submit your answer in two separate files as follow:**

- source code in a plain **text** format (**.txt**) (not screenshot/image object).  Please indicate what language is used in your HW and do NOT include line number in your source code.
- a screenshot in **PDF** format showing the execution instructions and execution result. Please indicate where the **online compiler** (e.g., **https://replit.com/languages/fsharp** for F#) can be used to run your code.

**PL: ch15**

**Homework Question**: **40** pts

Write a function in any **functional** programming language, e.g., Haskell, Scheme, or F#, that will **reverse** a **general** list.

For example, if an input data is (A (B C (D E)) F), the output will be (F ((E D) C B) A).  Please note that any built-in/pre-defined function, e.g., reverse, **cannot** be used in your answer.

Your code should work with any input data, not just for the sample data (A (B C (D E)) F).

Please DO NOT hard-code any input values, output values in your code.

Steps for execution

1. Copy the code from *"Advanced Software Paradigms_CSCI_6221_10_Sagar_Sheth_HW7.txt"*.
2. Open the below link in the web browser.
   a. https://www.tutorialspoint.com/compile_haskell_online.php
3. Paste the Code in and Click on the Execute Button.



```haskell
main :: IO ()
main = do
    putStrLn "Enter a string:"
    input <- getLine
    let result = map (\c -> if c == '(' then ')' else if c == ')' then '(' else c) input
    let result2 = map (\c -> if c == '{' then '}' else if c == '}' then '{' else c) result
    let result3 = map (\c -> if c == '[' then ']' else if c == ']' then '[' else c) result2
    let reversed = reverseString result3
    putStrLn ("Result: " ++ reversed)

reverseString :: String -> String
reverseString [] = [] -- Base case: empty string
reverseString (x:xs) = reverseString xs ++ [x]
```

4. The below output should be displayed on the terminal.

```
Terminal
[1 of 1] Compiling Main          ( main.hs, main.o )
Linking main ...
Enter a string:
 (A (B C (D E)) F)
 Result: (F ((E D) C B) A)
```

```
Terminal
[1 of 1] Compiling Main          ( main.hs, main.o )
Linking main ...
Enter a string:
{[(F ((E D) C B) A)]}
Result: {[(A (B C (D E)) F)]}
```