**CSCI 2315**

**Fall 2021**

**Assignment 4**

**Due: Tuesday October 26, 2021 by 11:59PM**

In this assignment you are to utilize the stack data structure to perform a function that you are familiar with when using Microsoft Word. Many times, you utilize the function “Undo” and “Redo” in Word to undo an action that you took, or redo something that you have already undone. If you push undo multiple times, then it will undo the actions that you performed in the reverse order in which you have done them. Similarly it will perform the redo operations by performed a bunch of undo the undo’s. You can see how a stack data structure is now utilized in this assignment. You must use the stack data structure that is provided in the book. The list-based stack that was provided for you.

What you need to do is to start a program by typing an arbitrary sentence with an arbitrary number of words. Once the sentence is entered, you can perform actions on the sentence. Your program should allow for any number of actions to be performed in the order that they appear. The following actions are allowed:

* add “some word”
* delete the previous word added
* undo the previous action
* redo a prior undo
* print the sentence (This is the only action that cannot be undone or redone)
  + This action should print the sentence after applying any previous actions mentioned before the call to the print.
* quit to terminate the program

These first five actions can actually be listed in any combination, and any one of them can appear multiple times after the first initial sentence is entered (Do not worry about any punctuations, only words and numbers as strings will be entered).

Implementation Details:

1. Once the sentence is entered, you should store it in any data structure of your choice. Array, Linked List, it’s up to you.
2. Upon typing an action, you should perform it on the sentence and then push that action on a stack (only push actions add and delete).
3. Upon doing an undo, you should take the action off the stack, reverse its action on the sentence and push it on a redo stack, so you can redo the undo.
4. If you issue an add or a delete, your redo stack should become empty. Basically, your redo stack can only have items after an undo operation is issued. Once you add or delete, you can’t redo, but you can undo as many operations as you have performed so far.

Example Scenario: A sentence with 5 words is entered

Please Enter a sentence: **Hello welcome to assignment 4**

**add I**

**add am**

**add going**

**add to**

**add finish**

**add it**

**add quickly**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it quickly**

**undo**

**undo**

**undo**

**print**

**Sentence: Hello welcome to assignment 4 I am going to**

**redo**

**redo**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it**

**add today**

**add I**

**add think**

**undo**

**undo**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it today**

**redo**

**redo**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it today I think**

**delete**

**delete**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it today**

**undo**

**undo**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it today I think**

**redo**

**print**

**Sentence: Hello welcome to assignment 4 I am going to finish it today I**

**quit**