CPSC 2430 Spring 2020 Programming Assignment #2

Due date: May 4 2020, 11:59pm

DO NOT USE any of the STL (including the vector or list type) in your code.

Recursive Stack

Implement a Stack ADT using a singly linked list. The class should be named RecursiveStack. Implement all basic class functions (*default constructor, copy constructor, destructor, copy assignment*) and basic stack functions (*pop, push, top, isEmpty*). Pop should be done from the head of the list.

Recursive Functions

Implement the following functions:

1. *recInsert(int value)*: a recursive function that takes in a positive integer and inserts each digit into the linked list.

If you run *recInsert(13204)* on an empty stack, it should create the following stack.

2. *printStars()*: a recursive function that prints each digit on the stack on a separate line as *'s. 0 is printed as "-". You can use a for loop to print the *'s.

The stack above will be printed as:

```
*
***
**
-
****
```

3. *print()*: a function that prints the stack with comma separated values. For example, the stack above will be printed as: 1, 3, 2, 0, 4
Note that there is no trailing comma. You can choose whether to implement it recursively or iteratively.

For printing functions (2 & 3), you can use a helper function that takes (*Node* head*) as a parameter.

Testing and submission

Provide a driver/client program to demonstrate your functions. You should ask the user for a number to pass to *recInsert*. Then call the function on an empty stack. After inserting nodes, you should call *print* and *printStars* on the resulting stack. You should also test all other functions of RecursiveStack class thoroughly.

Submit your program using the following command:
/home/fac/hkong/submit/cpsc2430/submit_pa2

The RecursiveStack class should be in two files named **recursiveStack.h** and **recursiveStack.cpp**. The driver should be in a file named **pa2.cpp**.

Execution example

```
[PA2]$ ./pa2
Welcome. Please enter a positive integer: 12345
Elements on the stack are:
1,2,3,4,5


Star print:
*
**
***
****
*****


[PA2]$ ./pa2
Welcome. Please enter a positive integer: 2019
Elements on the stack are:
2,0,1,9


Star print:
**
-
*
*********

Pushing 4
Elements on the stack are:
4,2,0,1,9
```