IIT PALAKKAD

COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology, Palakkad
CS2130: Data Structures and Algorithms Lab
Lab 5 (Stack using Linked List)

7 Sep, 2018

Time: 3 hrs

General Instructions

- You are expected to follow good programming practices (refer to the book titled *C How to Program* by Dietel and Dietel). Your programs should handle all kinds of test cases - positive, negative, corner cases/boundary conditions, good input, bad input and are never expected to crash on any input.
- You are also expected to provided a high level description of the functions used in your programs. There is no restriction on the format except that it should be concise and precise. Further, the running time and additional space (excluding the space for the arguments) used by the functions should be clearly stated and justified.
- **Please note that poor programming practices and improper documentation can attract a penalty of up to 20% of the marks allocated to a question.**

1 Part 1: Stack using array

1. Write code to implement a modified Stack ADT using arrays that support the following operations [30]

   Input-Output Format: The first line contains the value of n_1, the maximum number of non-negative integers that the stack can store. The second line contains the value of n_2, the maximum number of negative integers that the stack should store. The subsequent lines will contain specific operations on the stack.

   PSH x Insert the integer x into the stack. Output -1 if the operation was unsuccessful.
   POPN Delete the topmost negative integer from the stack. Output the value of the deleted integer if the operation was successful and 0 otherwise.
   POPP Delete the topmost non-negative integer from the stack. Output the value of the deleted integer if the operation was successful and -1 otherwise.
   PRTN Output the negative integers of stack in the reverse order in which they were inserted, one integer per line.
   PRTP Output the non-negative integers of stack in the reverse order in which they were inserted, one integer per line.

   ```
   Sample Input:
   5
   4
   PSH -4
   PSH 3
   PSH 0
   ```

Continued on next page...

```
PSH -5
PSH 7
PSH -8
PSH -2
POPN
PRTP
POPP
PRTN

Expected Output:
-2
7
0
3
7
-8
-5
-4
```

2 Part 2: Stack using singly linked list

1. Write code to implement the Stack ADT using singly linked list. The elements in the stack should have character data. [40]

   Input-Output Format: Each line contains a specific operation to be performed on the stack.
   PSH x denotes push(.,.). Insert the element x into the stack.
   POP denotes pop(.). Delete the topmost element from the stack. Output the value of the deleted element if the operation was successful and -1 otherwise.
   TOP denotes peek(.,.). Obtain the value of the topmost element from the stack. Return the value of this element if the operation is successful and -1 otherwise.
   PRT denotes print(.). Output the elements of stack in the reverse order in which they were inserted, one element per line.
   SZE denotes size(.). Output the number of elements in the stack.
   EMP denotes isEmpty(.). Return 1 if the stack is empty and 0 otherwise.

   ```
   Sample Input:
   PSH a
   TOP
   SZE
   PSH b
   PRT
   POP
   POP
   EMP

   Expected Output:
   a
   ```

Continued on next page...

```
1
b
a
b
a
1
```

2. Write a program that will take as input a string of parentheses and decide whether the symbols are balanced. Use the implementation of stack using linked list of the previous question for storing and processing the input expression. [30]

   Input-Output Format: The input consists of a single line containing the input expression. Each character in the expression is an element in $\{\{, \}, (,), [,]\}$. The output is 1 if the expression is balanced and 0 otherwise.

   ```
   Sample Input:
   (){((()))[()]}

   Expected Output:
   1
   ```