IIT PALAKKAD

COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology, Palakkad

# CS2130: Data Structures and Algorithms Lab

***Lab 6 (Application of Stack)***

14 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: Evaluation of Postfix Expression

1. Recall that in the postfix form of an arithmetic expression involving only binary operators, the operator follows the second operand which in turn follows the first operand. For example, the postfix form of $x/y$ is $xy/$. Write a program that will take as input an arithmetic expression in postfix form and evaluate it. Assume that each operator is a binary operator from the set $\{+, -, *, /\}$ and that each of the operands is an integer. Also, assume that / is the integer division operator. [30]

   **Input-Output Format:** The input consists of the input postfix expression specified as multiple lines. Each line either contains an integer or an operator from the set $\{+, -, *, /\}$. If the input expression is a valid postfix expression, then the output is a single line containing the value of that expression followed by the newline character. Otherwise, the output is the string "invalid" followed by the newline character. If the input expression is valid but a division operation is attempted with zero as the divisor during the evaluation, the output has to be "division by 0" followed by the newline character.

```
Sample Input 1:
1
2
3
27
*
*
+
```

Continued on next page. . .

```
Expected Output:
163

Sample Input 2:
1
+

Expected Output:
invalid

Sample Input 3:

Expected Output:
invalid

Sample Input 4:
5
10
4
6
+
-
/

Expected Output:
division by 0

Sample Input 5:
5
10
4
6
+
-

Expected Output:
invalid
```

## 2 Part 2: Conversion of Infix Form to Postfix Form

1. Write a program that will take as input a **valid** arithmetic expression in infix form and covert it into postfix form. Assume that each operand is represented by a single lower case letter and that each of the operators come from the set $\{+, -, *, /\}$. Assume the BODMAS rule with the following standard precedence of the operators: (i) $*$ and $/$ have equal precedence (ii) $+$ and $-$ have equal precedence (iii) any operator from the set $\{+, -\}$ has a lower precedence than

Question 1 continues on the next page...

any operator in the set $\{/, *\}$. Further, each of these operators is left-associative. [70]

**Input-Output Format:** The input consists of a single line containing the input expression in infix form. Each character in the expression is either a letter or an element in $\{+, -, *, /, (, )\}$. The output is a single line containing the postfix form of the input expression followed by the newline character.

```
Sample Input:
a+b*(c-d)+(e+f*g)-h

Expected Output:
abcd-*+efg*++h-
```