Computing Systems Lab

IT69101

(Autumn 2010)

Assignment #2 28/07/2010

1. Write C programs for the following a stack data structure (of array representation).

(a) Evaluation of an arithmetic expression

Description: You are given an arithmetic expression in infix form, and the values that the variables hold. You have to evaluate the expression with the given values and print the output.

Input:
First line has a number T, that is number of test cases.
Then for each one of T pairs, first line has an arithmetic expression followed by values assigned to the variables used in the expression. Each test case is separated by two new lines.

Output:
Print the evaluated value for given input for each test case in separate line.

Sample Input:
3
a+b*c
2 1 3
(a+b)/(c+d)
4 2 1 2
a+b^(c*d)
0 2 1 4

Sample Output:
5
2
16

(b) Conversion of an expression (in infix) to postfix notation.

Description:
You are given an infix expression, which is to be converted in to postfix notation.

Input:
First line of the input contains number of test cases, T.
Then follows T lines with an infix expression in each line.

Output:
Corresponding T postfix expressions for given infix expressions in that order.

Sample Input:
3
a+b*c
(a+b)/(c+d)
a+b^(c*d)

Output:
abc*+

ab+cd+/
abcd*^+

(Your implementation should consider stack operations like PUSH, POP, TOP etc.)

2. Using linked list representation of stack data structure, solve the following in C programming language.

(a) Tower of Hanoi Problem

Description:
You are given N discs which are piled on a source rod (A), which are to be moved on to a target (C) , using an auxiliary rod (B). You have to show the transformations of the discs in the following manner.
DiscNumber: Rod1 -> Rod2

Input:
First line of the input contains number of test cases, T.
Next T lines contain number of Discs.
Output:
You have to print the transformation of discs in the specified format for each case.

Sample Input:
2
2
3

Output:
Disc1: A -> B
Disc 2: A -> C
Disc 1: B -> C

Disc1: A -> C
Disc2: A -> B
Disc1: C -> B
Disc3: A -> C
Disc1: B -> A
Disc2: B -> C
Disc1: A -> C

(b) Calculation of n-th Fibonacci number F_n, where $F_n = F_{n-1} + F_{n-2}, F_1 = 1, F_2 = 1$

Description:
You have to calculate the nth number in the Fibonacci series for the given n.

Input:
First line of the input contains number of test cases, T.
Next T lines contain number n.

Output:
Print nth Fibonacci number for each test case.

Sample Input:
4
2
5
3
1

Output:
1
5
2
1

(Your implementation should consider stack operations like PUSH, POP, TOP etc.)

3. Implement queue using a circular array. Test your implementation with the following.

i. Insertion into an empty queue
ii. Deletion from an empty queue
iii. Insertion into a full queue
iv. Deletion from a full queue
v. Insertion into a non-empty queue with space for only one item
vi. Insertion into a non-empty queue with space for more than one item
vii. Deletion from a non-empty queue with only one item in the queue
viii. Deletion from a non-empty queue with more than one item in the queue

Description:
For each operation assign a code as following.

1. Insert
2. Delete
3. Display
4. Exit

Input:
You are given sequence of operations and corresponding value if the operation is insert (1).

Output:
When ever you encounter Display operation (3), you should display the contents of the Queue.

Sample Input:

1 37
1 54
1 81
3
2
2
3
1 25
1 123
1 365
3
2
3
4

Output:
37 54 81
81
81 25 123 365
25 123 365

4. Suppose priority values for any element varies in between 1 and 5, both inclusive. Implement a priority queue using the following data structures.

(a) A single array of size n, $n >> 5$.

(b) A single linked list

(c) A 2D matrix.

Description:
For each operation assign a code as following.

1. Insert
2. Delete
3. Display
4. Exit

Input is same for all the sub problems 4a, 4b and 4c

Input:
You are given sequence of operations and corresponding values (actual value and priority) if the operation is insert (1).

Output:
When ever you encounter Display operation (3), you should display the contents of the Queue along with their priority.

Sample Input:

```
1 37 3
1 54 1
1 81 2
3
2
2
3
1 25 1
1 123 3
1 365 2
1 50 1
3
2
3
2
3
4
```

Output:

```
37(3) 54(1) 81(2)
37(3)
37(3) 25(1) 123(3) 365(2) 50(1)
37(3) 123(3) 365(2) 50(1)
37(3) 123(3) 365(2)
```

Last date of submission: 04/08/2010