CS 14 Summer 2016: Exam # 3

Name: Apollo Truong

Last 4 Student ID 8740

1. Evaluate the following postfix expression using the algorithm presented in class. Show the stack and output after each step.

2 7 8 12 + 8 \* - 5 + \*

|  |  |
| --- | --- |
| Current Symbol | Stack |
| 2 | 2 |
| 7 | 2 7 |
| 8 | 2 7 8 |
| 12 | 2 7 8 12 |
| + | 2 7 20 |
| 8 | 2 7 20 8 |
| \* | 2 7 160 |
| - | 2 -153 |
| 5 | 2 -153 5 |
| + | 2 -148 |
| \* | -296 |

1. What does the postfix expression in question 1 evaluate to?

-296

1. Give the psuedocode for the stack pop algorithm as presented in class.  Algorithm pop():
   * + 1. if is\_empty() then
          1. throw stack\_empty\_exception

t <- t - 1

1. Give the pseudocode for the stack push algorithm as presented in class. if size() == N then

Throw stack\_full\_exception

t <- t + 1

S[t] <- o

1. What type of error can occur on a stack push operation? Assume the stack is array-based.

The error that can occur is a stack\_overflow error. When you push data on an already full stack.

6. Convert the following infix expression to postfix notation using the algorithm presented in class. Show the stack and output after each step.

2 + 4 \* 8 - (6 \* 2 + 12)

|  |  |
| --- | --- |
| Current Symbol | Stack |
| 2 | 2 |
| + | 2 4 |
| \* + | 2 4 8 |
| \* + - | 2 4 8 \* + |
| - ( | 2 4 8 \* + 6 |
| -(\* | 2 4 8 \* + 6 2 |
| -(+ | 2 4 8 \* + 6 2 \* 12 |
| - | 2 4 8 \* + 6 2 \* 12 + |
|  | 2 4 8 \* + 6 2 \* 12 + - |

1. What postfix expression is produced in question 6?

2 4 8 \* + 6 2 \* 12 + -

1. What is the worst-case running time in O-notation for removing an item from a stack class implemented using a linked list?

O(1)

1. Show an array-based Stack object S after the following operation sequence. Assume that S is empty and has size 5.

 S.push(5), S.top(), S.push(6), S.push(7), S.pop().

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 5 |  |  |  |

10. Convert the following function into a function template.

// returns the min of a and b int min(int a, int b){

return (a < b ? a : b); }

template<class T>

T min(T a, T b){

return (a < b ? a : b);

}