

CS311 Yoshii - Homework 1 Part 1 Getting Started (based on Weeks 1-2)

Topics: Stack and Queue application

Due: Week 3 Friday

Total: 42 points. Your score is:

Your Name: Eduardo Martinez

Date Submitted: 8/16/16

Purpose: To be able to choose the correct data structure for each problem.

For all HW: Must test your program on empress with g++.
YOU MUST answer the State of the Program questions honestly or you will lose points. They will help us debug your program.

Q0) Initial State of the Program Statement [4pts]:

1. Does your stack class compile without errors? If not, describe. yes
2. Does it run with stacktest.cpp without errors? yes
3. Does your queue class compile without errors? If not, describe. yes
4. Does it run with queuetestp.cpp without errors? yes

Problem 1: Write a stack application program to evaluate post-fix expressions.
[8+12=20pts] Your score is:

Must use my stack.h, stack.cpp, client1.cpp templates. Run my solution program first to study the output.

The user will enter a post-fix expression of the form

34+ which means 3+4
345+* which means 3*(4+5)
722+- which means 7-(2+2)

as a string. Operators are +, -, and *. Single digit numbers only. No blanks.

Please note that the expression could be wrong.

e.g.

- 3+ - too few operands (i.e. cannot pop operand)
345 - incomplete expression (i.e. more than the result at the end)

Your program will display the evaluated result (a number) or an error message describing what is wrong (the algorithm is in Notes-1, so use it as given).

- You may assume that no expression will be longer than 12 characters.
- Since each element of the string is a character, you will need to convert it to an integer to perform arithmetic operations.

Q1) Warm Up Exercises [4pts = 1 pt per question]:

For each, report the type of error (and why the error occurred) and/or the computed result.

Do not say Overflow or Underflow!

Input1: 3141-*+5- = 1

Input2: 314-*+5- too few operands

Input3: 314111111.. (beyond the size of the stack) expression too long

Input4: 3141+* incomplete expression

Q2) [2pts]

- What data structure do you plan to use – integer stack or character stack?
integer stack
- Why?
we are using numbers

Now, implement your client program.

Required Test Cases (MUST TEST IN THIS ORDER):

- | | |
|------------------|-------------------------------------|
| 1. 34+ | which means 3+4 |
| 2. 345+* | which means 3*(4+5) |
| 3. 722+- | which means 7-(2+2) |
| 4. 34+56++ | which means (3+4)+(5+6) |
| 5. 12+34*45+-+ | which means (1+2) + ((3*4) - (4+5)) |
| 6. 1234567891234 | expression too long |
| 7. + | too few operands |
| 8. 3+ | too few operands |
| 9. 3# | invalid element |
| 10. 2345+ | incomplete expression |

NOTE: Do you think I have listed all possible cases? Always ask yourself if there are other cases you should test to make sure your program is bug free.

Q3) State of the Program Statement [2pts]:

- Does your program compile without errors? If not, describe. yes
- List any run-time bugs you are aware of, or state “No bugs”: No Bugs

Problem 2: Write a C++ program to generate all strings using A, B, and C as the letters. [6+12=18pts] Your score is:

Must use my_queue.h queue.cpp, client2.cpp templates. Run my solution program first to study the output.

The strings must be generated in the following order:

A

B
C
AA
AB
AC
BA
BB
BC
CA
CB
CC
AAA
AAB
AAC
ABA
ABB
ABC
ACA
ACB
ACC
etc.

First answer these questions:

Q4) What data structure do you plan to use (char queue or string queue)?[2pts]
char queue

Q5) Describe how this data structure will allow you to generate the strings in the above order. Give the algorithm in English. Demonstrate using an example. [2pts]
By using character it allows us to add them together. for example 'a' + 'a' will equal to 'aa'.

Implement the client program. The program should keep on going until your data structure overflows. At that point, at least 25 strings should have been displayed. Then, answer the following:

Q6) State of the Program Statement [2pts]:

- Does your program compile without errors? If not, describe. yes
- List any bugs you are aware of, or state "No bugs": no bugs

SUBMIT THESE 9 FILES TO COUGAR COURSES:

Always make sure the files you submit can be opened on a PC and be edited

1. This file with your answers inserted (docx, doc, or rtf)
2. Program 1

- Stack.h Header file with good comments.
- Stack.cpp Implementation file with good comments.
- Client1.cpp Client program with good comments.
- Test1 Script result of compiling & executing your program on
empress.csusm.edu for the required test cases. Submit even if your
program does not compile.

3. Program 2

- Queue.h Header file with good comments.
 - Queue.cpp Implementation files with good comments.
 - Client2.cpp Client program with good comments.³⁴
 - Test2 Script result of compiling & executing your program on
empress.csusm.edu. Submit even if your program does not compile.
-
- Whether working or not, test result must include the lines for compiling your files or we will not grade your program i.e. 0 points for the program.
 - Did you check your comments and style against CS311 How To Comment.doc??
 - Did you answer all the questions?