

Programming Assignment #7

Match Brackets using a Stack

CS 2308.255 and 256, Spring 2013

Instructor: Jill Seaman

Due:

section 255: in class **Tuesday, 4/23/2013** (upload electronic copy by 4:00pm)

section 256: in class **Monday, 4/22/2013** (upload electronic copy by 1:00pm)

Problem:

Given a text file, your program will determine if all the parentheses, curly braces, and square brackets match, and are nested appropriately. Your program should work for mathematical formulas and computer programs.

Your program should read in the characters from the file, but ignore all characters except for the following:

{ } () []

Your program should use the IntStack implementation (IntStack.h and IntStack.cpp, which will be available from the class webpage).

The general algorithm is to use the stack to store the opening unmatched brackets. When a closing bracket is encountered, check it against the one on top of the stack (pop it off)--make sure it matches. When you are finished there should be no unmatched brackets left on the stack.

Input/Output:

Your program should prompt the user to enter a filename. It should then try to open the file and then check it make sure the brackets all match appropriately. If they all match, the program should output a message saying so. If not, the program should output an appropriate error message.

There are three types of errors that can happen (and they can happen with any kind of bracket):

missing } : if you reach the end of the file, and there is an opening { that was never matched, like: `int main () { x[size]=10;`

expected } but found) : this is a wrong closing bracket, like: `{x[i]=10;)...`

unmatched } : this occurs if there is a closing bracket but not an opening bracket (not even one of the wrong kind), like: `int main () { x[i]=10; } }...`

NOTES:

- The stack is an int stack but you will probably want to put characters on the stack. You can push a character on the int stack, but you must pop into an int variable. This is only a problem if you want to display the popped character to the screen. If so you will need to convert it to a char first.
- It might be easier to store the expected closing character on the stack when an opening bracket is encountered. This simplifies the matching when a closing bracket is encountered.
- Follow the style guidelines from the class website.

Logistics:

For this assignment you will have only one file to submit, the one containing your main function. Name your file **assign7_XXXXXX.cpp** (where XXXXX is your NetID)..

Submit an electronic copy using the Assignments tool in TRACS no later than one hour before class the day the assignment is due (see top of page 1).

For this assignment, no printout is required! The feedback cover sheet will be posted to the assignments tool in TRACS after the assignments are graded (and the grade will be posted to the TRACS gradebook2 tool).