

CS 211 – Project 2

Discussion of Project

Matching Symbol Pairs

Distinct Character for Opening Symbol and Closing Symbol in each pair

- Parentheses:

- Opening: (Closing:)

- Curly Braces:

- Opening: { Closing: }

- Square Brackets:

- Opening: [Closing:]

- Angle Brackets:

- Opening: < Closing: >

Matching Symbol Pairs

Must have same number of Opening Symbols as Closing Symbols of each pair type

- Valid: (([]))
- Invalid: (()

Proper Nesting of symbol pairs is required

- Valid: < { } >
- Invalid: < { > }

Opening Symbol must occur before paired Closing Symbol

- Valid: [< >]
- Invalid: } > < {

Three Errors to Report

Error 1: Expecting a different Closing Symbol

- (([]])
 ^ expecting)

Error 2: Missing an Open Symbol (i.e. extra closing symbol)

- ({ })]
 ^ missing [

Error 3: Missing a Closing Symbol (i.e. expression ends too early)

- < { () }
 ^ missing >

Three Errors to Report

You must specify:

- The location of the error (assume a fixed-width font for output)
- The “type” of error
- The proper symbol that should solve the error

If a line contains multiple errors, **ONLY REPORT THE FIRST ERROR**

- Stop processing that line once the first error is found

General Algorithm

When you encounter an opening symbol:

- Push the symbol onto the stack

When you encounter a closing symbol:

- Verify the top of the stack contains the matching opening symbol
- Pop/remove the opening symbol from the stack

When you encounter the end of the expression

- Verify the stack is empty => Expression is Balanced

Ignore any other symbols contained in the expression

General Algorithm Error Checks

When you encounter an opening symbol:

- No error checking done here

When you encounter a closing symbol:

- If the stack is empty => Error # 2
- Non-matching opening symbol on top of stack => Error # 1

When you encounter the end of the expression

- If stack is not empty => Error #3

Example Valid Expression

Expression: < [a] >

Current Symbol: At start of Expression

Symbol Type: Nothing Yet

Task: Clear stack



Stack

Example Valid Expression

Expression: < [a] >

^

Current Symbol: <

Symbol Type: Opening

Task: Push onto stack



Stack

Example Valid Expression

Expression: < [a] >

^

Current Symbol: <

Symbol Type: Opening

Task: Push onto stack



Stack

Example Valid Expression

Expression: < [a] >
 ^

Current Symbol: [

Symbol Type: Opening

Task: Push onto stack



Stack

Example Valid Expression

Expression: < [a] >
 ^

Current Symbol: [

Symbol Type: Opening

Task: Push onto stack



Stack

Example Valid Expression

Expression: < [a] >
 ^

Current Symbol: a

Symbol Type: None

Task: ignore



Stack

Example Valid Expression

Expression: < [a] >

^

Current Symbol: At End of Expression

Symbol Type: End of Expression

Task: Verify stack is Empty



Stack

Example Valid Expression

Expression: < [a] >

^

Current Symbol: At End of Expression

Symbol Type: End of Expression

Task: Verify stack is Empty

Stack is EMPTY, so Expression is Balanced!



Stack