Fundamentals of
Data Structures using C

Balancing Symbols

B.Bhuvaneswaran, AP (SG) / CSE

☎ 9791519152
✉ bhuvaneswaran@rajalakshmi.edu.in

RAJALAKSHMI
ENGINEERING COLLEGE

Introduction

- Compilers check your programs for syntax errors, but frequently a lack of one symbol (such as a missing brace or comment starter) will cause the compiler to spill out a hundred lines of diagnostics without identifying the real error.

Introduction

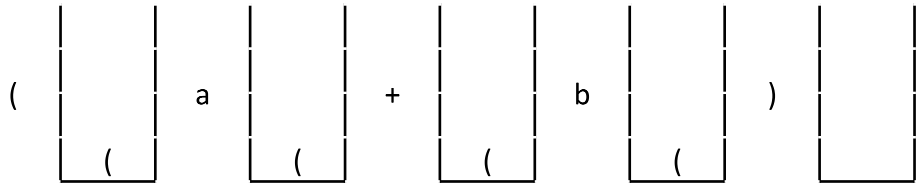
- A useful tool in this situation is a program that checks whether everything is balanced.
- Thus, every right brace, bracket, and parenthesis must correspond to their left counterparts.
- The sequence [()] is legal, but [(]) is wrong.
- Obviously, it is not worthwhile writing a huge program for this, but it turns out that it is easy to check these things.
- For simplicity, we will just check for balancing of parentheses, brackets, and braces and ignore any other character that appears.

Algorithm

- Make an empty stack.
- Read characters until end of input.
- If the character read is not a symbol to be balanced, ignore it.
- If the character is an opening symbol like (, [, {, push it onto the stack.
- If it is a closing symbol like),], }, then if the stack is empty report an error as "Missing opening symbol". Otherwise, pop the stack.
- If the symbol popped is not the corresponding opening symbol, then report an error as "Mismatched symbol".
- At end of input, if the stack is not empty report an error as "Missing closing symbol". Otherwise, report as "Symbols are balanced".

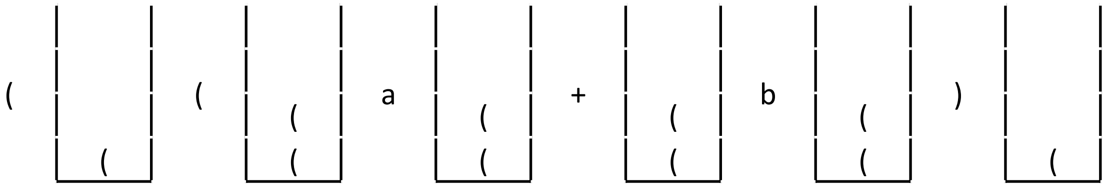
(a+b)

- Stack is empty → Symbols are balanced.



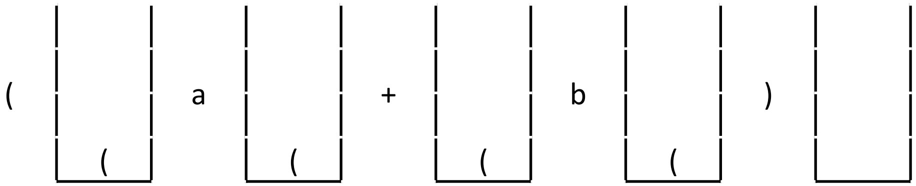
((a+b)

- Stack is not empty → Missing closing symbol.



(a+b))

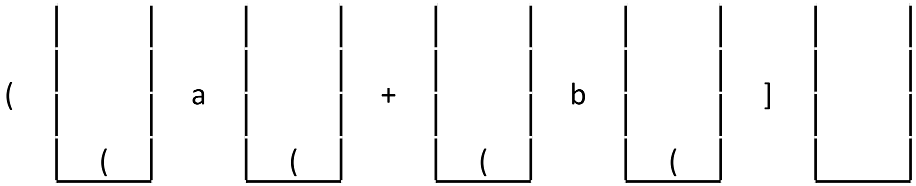
- Stack is empty → Missing opening symbol.



((a (+ (b ( )

(a+b]

- Not corresponding to the opening symbol → Mismatched symbol.



Queries?

Thank You!