Lab Assignment: Application of Stack - Checking for Balanced Parentheses

Problem Statement

Given an expression containing characters like (,), {, }, [,], determine whether the expression is balanced. An expression is considered balanced if:
- Every opening bracket has a corresponding closing bracket.
- Brackets are closed in the correct order.

Example

- {[()]} → Balanced
- (()() → Not Balanced
- {[(])} → Not Balanced

Algorithm

1. Initialize an empty stack.
2. Traverse each character in the expression.
3. If the character is an opening bracket, push it onto the stack.
4. If it's a closing bracket:
5. - If the stack is empty, return false (unbalanced).
6. - If the top of the stack does not match the corresponding opening bracket, return false.
7. - Otherwise, pop the opening bracket from the stack.
8. After processing, if the stack is empty, the expression is balanced.

Function Signature and Description

You are required to implement the following functions for the assignment:

- int isMatchingPair(char open, char close)
  → Returns true if the pair of brackets match.

- int isBalanced(char* expr)
  → Returns 1 if the given expression is balanced, otherwise 0.

Sample Input and Output

Input: {[()]}

Output: Balanced

Input: [(])

Output: Not Balanced