



- 初始化：
 - 我們使用一個 `top` 變數來模擬堆疊的高度，堆疊用來存儲尚未配對的開括號。
 - `input[200]` 用來存儲每一行的輸入，並且每行最多 200 個字符。
 - 使用 `s[100]` 作為堆疊，存儲開括號。
- 迴圈開始：
 - 程式透過 `cin.getline(input, 200)` 來讀取一行輸入。這會將一行字符串存儲在 `input` 中。
 - 每讀取一行後，將堆疊指標 `top` 設為 0，表示新一輪的括號配對檢查。
 - `isValid` 變數用來檢查括號配對過程是否有效（即沒有發現錯誤的配對情況）。
 - `push` 用來檢查是否有進行過括號的推入動作，這有助於處理空字符串或只有閉括號的情況。
- 檢查每個字符：
 - 使用 `for` 迴圈來逐一檢查 `input` 字符串中的每個字符。

- **處理開括號：**
 - 當遇到開括號 (、{ 或 [時，將其推入堆疊 $s[top]$ ，並將 top 加 1，表示堆疊上升。push 標記為 true，表示有開括號被推入堆疊。
- **處理閉括號：**
 - 當遇到閉括號)、} 或] 時，檢查堆疊的頂端是否有對應的開括號。
 - 如果 $top > 0$ 且堆疊頂端的括號是對應的開括號，則將堆疊指標 top 減少，表示已經配對成功，並將這個開括號從堆疊中移除。
 - 如果堆疊中沒有對應的開括號，則設置 isValid 為 false，表示括號配對不正確。
- **結果判斷：**
 - 如果整行處理完後， top 等於 0 且 isValid 為 true，表示所有括號都成功配對，程式輸出 true。
 - 如果 top 不為 0，表示有開括號沒有匹配的閉括號，或者在處理過程中發現了無效的配對，則程式輸出 false。
 - 若 push 標記為 false（即輸入字符串中沒有括號），程式不輸出任何東西。