|  |  |
| --- | --- |
| **Balanced Parenthesis in C++** | |
| #include <iostream>  #include <stack>  using namespace std;  bool isBal(string str) {  stack<char> s;  for (int i = 0; i < str.length(); i++) {  if (str[i] == '(' || str[i] == '{' || str[i] == '[') {  s.push(str[i]);  } else {  if (s.empty()) {  return false;  } else if ((str[i] == ')' && s.top() == '(') || (str[i] == '}' && s.top() == '{') || (str[i] == ']' && s.top() == '[')) {  s.pop();  } else {  return false;  }  }  }  return s.empty();  }  int main() {  cout << boolalpha << isBal("(())") << endl; // Example usage  return 0;  } | **Function Purpose**  Checks if the string contains balanced brackets:   * (), {}, and []   **🔍 Input**  string str = "(())"  **🔧 Stack Simulation Table**   | **i** | **str[i]** | **Stack Before** | **Action** | **Stack After** | | --- | --- | --- | --- | --- | | 0 | ( | [] | Push '(' | ['('] | | 1 | ( | ['('] | Push '(' | ['(', '('] | | 2 | ) | ['(', '('] | Top '(' matches ) → Pop | ['('] | | 3 | ) | ['('] | Top '(' matches ) → Pop | [] |   **✅ Final Check:**   * Stack is empty → Balanced * Output: true   **💡 Output:**  true |
| true | |