Its Your Classic Bracket!

Problem Submissions Discussions

Hi there! My name is Ciko, and I'm organizing a set of boxes represented by different types of brackets. The brackets I have are: `{}`, `[]`, `()`, and `<>`. Each type of bracket has a specific strength or power level, indicating the order they need to be closed.

Your task is to determine the validity of a given string that represents the arrangement of these brackets. To be considered valid, the following conditions must be satisfied:

1. Every opening bracket must have a corresponding closing bracket of the same type.
2. Brackets must be closed in the correct order based on their strength:
   - The `{}` brackets have a strength of 1.
   - The `[]` brackets have a strength of 2.
   - The `()` brackets have a strength of 3.
   - The `<>` brackets have a strength of 4.
3. The closing of brackets must follow the correct order of strength.

Input Format

One line of string that contains bracket of `{}`,`[]`,`()`,`<>`

Constraints

Only the brackets, lol

Output Format

If the conditions fulfilled, print valid. Otherwise, print invalid

Sample Input 0

```
{[()]}
```

Sample Output 0

```
Valid
```

Explanation 0

The arrangement `{[()]}` is valid because each opening bracket has a corresponding closing bracket of the same type, and they are closed in the correct order of strength.

Sample Input 1

```
{[()<>]}
```

Sample Output 1

Alt+Q

```
Invalid
```

Explanation 1

- The arrangement `{[()]}` is valid because each opening bracket has a corresponding closing bracket of the same type, and they are closed in the correct order of strength.
- On the other hand, the arrangement `{[()<>]}` is invalid because the closing bracket `]` does not match the opening bracket `[` in terms of type and strength.

Contest ends in 1 hour 10 minutes 1 second

Submissions: 6
Max Score: 1

Rate This Challenge:

More

C

```cpp
// C++ program to check for balanced brackets.

#include <iostream>
#include <stack>
using namespace std;
int indicator = 0;

// Function to check if brackets are balanced
bool areBracketsBalanced(string expr)
{
    // Declare a stack to hold the previous brackets.
    stack<char> temp;
    for (int i = 0; i < expr.length(); i++) {
        int tempInd = 0;
        if (temp.empty()) {

            // If the stack is empty
            // just push the current bracket
            temp.push(expr[i]);
        }
        else if ((temp.top() == '(' && expr[i] == ')')
                || (temp.top() == '{' && expr[i] == '}')
                || (temp.top() == '[' && expr[i] == ']')
                || (temp.top() == '<' && expr[i] == '>')) {

            // If we found any complete pair of bracket
            // then pop
            temp.pop();
            switch (expr[i])
                {
                case '}':
                    tempInd = 1;
                    break;
                case ']':
                    tempInd = 2;
                    break;
                case ')':
                    tempInd = 3;
                    break;
                case '>':
                    tempInd = 4;
                    break;

                default:
                    break;
                }
            if(indicator == 0){
```

```
                indicator = tempInd;
            }
            else if (indicator != 0){
                if(tempInd++ == indicator || tempInd-- == indicator){
                    indicator = tempInd;
                }
                else{
                    return false;
                }
            }
        }
        else {
            temp.push(expr[i]);
        }
    }
    if (temp.empty()) {

        // If stack is empty return true
        return true;
    }
    return false;
}

// Driver code
int main()
{
    string expr;
    cin >> expr;

    // Function call
    if (areBracketsBalanced(expr))
        cout << "Valid";
    else
        cout << "Invalid";
    return 0;
}
```

Line: 83 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

Compile time error

Compile Message

```
Solution.c:3:10: fatal error: iostream: No such file or directory
 #include <iostream>
          ^~~~~~~~~~
compilation terminated.
```

Exit Status

```
1
```