The Wizard's Lockbox

Story:

In the heart of Eldoria, the legendary (and slightly mischievous) wizard Zaruk has hidden his greatest treasures in a magical lockbox, protected by an enchanted sequence of magical symbols. Only those who can decipher the puzzle of these symbols can unlock the box. However, Zaruk, being a wizard with a twisted sense of humor, has added layers of protection by introducing decoy symbols and one-time use keys, just to mess with anyone who dares approach.

Your task is to help a brave adventurer unlock the treasure by validating the sequence of magical symbols. But be warned: if you fail, the lockbox will magically trap you in an eternal riddle competition with Zaruk—who is rumored to have a terrible sense of humor.

Legend has it, if you ask Zaruk to tell a joke, he'll respond:

"Why don't wizards trust stairs? Because they're always up to something!"

And that's why it's better to solve the puzzle.

Challenge:

The sequence is composed of:

1) Magical symbols used for unlocking:

Round Symbols: ()

Curly Symbols: {}

Square Symbols: []

Angle Symbols: < > (One-time use keys)

2) Decoy symbols that do not impact the logic of unlocking:

*, #, and @

Rules for Validating the Sequence

- The round, curly, and square symbols must be properly nested and balanced. That is, each opening symbol must have a corresponding closing symbol of the same type, and they must be closed in the correct order.
- The angle symbols are special one-time use keys. They must open and close immediately without any other magical symbols inside them. For example, "<>" is valid, but "<[()]>" or "<{}>" is not allowed.
- The decoy symbols (*, #, @) can appear anywhere in the sequence and must be ignored when checking the validity of the magical symbols.
- A valid sequence must have all magical symbols balanced, the one-time use keys (<>) must be used correctly, and the decoy symbols should not interfere with the balancing.

Input:

• A string sequence containing round, curly, square, and angle symbols as well as the decoy symbols.

Output:

• Print **true** if the sequence is valid and can unlock the wizard's lockbox; otherwise, print **false**.

Sample:

No.	Sample Input	Sample Output	Explanation
1	{[()]*}	true	Properly nested with a decoy *
2	[*(])	false	Mismatched closing square bracket, even though * is a decoy
3	{@#}*	true	Properly nested symbols; decoys @, #, and * are ignored.
4	{[<>]@}	true	Proper use of angle symbols < > with nothing nested inside them
5	(<*@>)	true	Decoy symbols are ignored
6	(<><>)()	false	Angle brackets used multiple times

Remember, every challenge you solve sharpens your mind and brings you one step closer to becoming a coding wizard - keep pushing forward, and the treasure of knowledge will be yours!