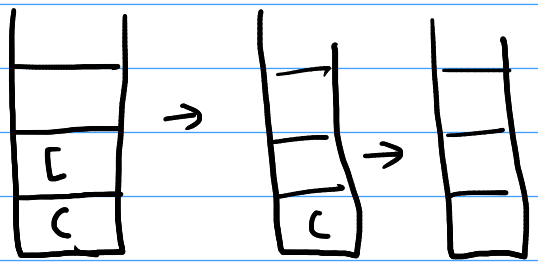
Valid Parantheses:

"([ ])" →




```
def check(input: str) -> bool:

    symbolStack = []
    balanced = False          # flag
    open_sym = ["(", "[", "{", "<"]
    close_sym = [")", "]", "}", ">"]

    for symbol in input:
        if symbol in open_sym:
            symbolStack.append(symbol)
        else:
            """ if the current symbol is NOT in open_sym,
            then it must be in close_sym. In such case,
            the stack must be empty after the
            operation. It CANNOT have any symbol
            left in it."""

'implies stack is not empty.' →  if not symbolStack:
                                     return False

            current_sym = symbolStack.pop()

            if current_sym == '(':
                if symbol != ')':
                    return False
```

```
if current_sym == "[":
    if symbol != "]":
        return False

if current_sym == "{":
    if symbol != "}":
        return False

if current_sym == "<":
    if symbol != ">":
        return False
```

```
# Now exit the loop, and check if the stack
# is empty or not.
# if empty → True, else → False

if symbolStack:
    return False
return True
```