```python
"""
CSE212
(c) BYU-Idaho
03-Teach - Problem 2 - Solution

It is a violation of BYU-Idaho Honor Code to post or share this code with others or
to post it online.  Storage into a personal and private repository (e.g. private
GitHub repository, unshared Google Drive folder) is acceptable.
"""

def check_braces(line):
    """
    This function will check a line of code an ensure that the braces (including parentheses,
    square brackets, and curly brackets) are properly used by matching and balancing the
    open and closed symbols.  All other characeters in the line of code will be ignored.
    A stack is used to remember which open braces where seen previously.  When a close
    brace is seen then the stack is checked to make sure there is a matching open brace.
    If an error is found, the function exits.
    """
    stack = []
    for item in line:     # Look at each character in the line of code received
        if item == '(' or item == '[' or item == '{':  # Check for open braces
            stack.append(item)
        elif item == ')':                               # Check for close braces
            if len(stack) == 0 or stack.pop() != '(':
                return False
        elif item == ']':
            if len(stack) == 0 or stack.pop() != '[':
                return False
        elif item == '}':
            if len(stack) == 0 or stack.pop() != '{':
                return False
    # If there were any open braces not yet closed then we have an error.
    return len(stack) == 0


# Good
print(check_braces("(a == 3 or (b == 5 and c == 6))"))

# Bad .... missing closing square bracket here -----------
#                                                      \/
print(check_braces("(students[i].grade > 80 and students[i.grade < 90)"))

# Bad .... missing closing ')' here -------
#                                      \/
print(check_braces("(robot[id + 1].execute(.pass() or (not robot[id * (2 + i)].alive
and stormy) or (robot[id - 1].alive and lava_flowing))"))
```