```python
class Stack_:
    def __init__(self):
        self.top = []

    def isEmpty(self): return len(self.top) == 0
    def size(self): return len(self.top)
    def clear(self): self.top = []

    def push(self, item):
        self.top.append(item)

    def pop(self):
        if not self.isEmpty():
            return self.top.pop(-1)

    def peek(self):
        if not self.isEmpty():
            return self.top[-1]

def checkBrackets(statement):
    n=0
    stack = Stack()
    for ch in statement:
        if ch in ('{', '[', '('):
            stack.push(ch)

        elif ch in ('}', ']', ')'):
            if stack.isEmpty():
                return False
            else:
                left = stack.pop()
                if (ch == "}" and left != "{") or \
                   (ch == "]" and left != "[") or \
                   (ch == ")" and left != "("):
                    return False
```

if checkBrackets(str) == False

```python
def checkBrackets(statement):
    n=0
    stack = Stack()
    for ch in statement:
        if ch in ('{', '[', '('):
            stack.push(ch)

        elif ch in ('}', ']', ')'):
            if stack.isEmpty():
                return False
            else:
                left = stack.pop()
                if (ch == '}' and left != '{') or \
                   (ch == ']' and left != '[') or \
                   (ch == ')' and left != '('):
                    return False

    return stack.isEmpty()

str = str(input("입력 :"))
n=0
for s in str:
    if (s=="(" or s == "{" or s == "[" or s == ")" or s== "}" or s == "]"):
        n+=1
m = checkBrackets(str)
if checkBrackets(str) == False:
    m = 'Wrong'
else:
    m = 'OK'

print(m,'_',n)
```

```python
class Stack_:
    def __init__(self):
        self.top = []

    def isEmpty(self): return len(self.top) == 0
    def size(self): return len(self.top)
    def clear(self): self.top = []

    def push(self, item):
        self.top.append(item)


    def pop(self):
        if not self.isEmpty():
            return self.top.pop(-1)

    def peek(self):
        if not self.isEmpty():
            return self.top[-1]


def checkBrackets(statement):
```

Run output:

```
/Users/cheonjiu/PycharmProjects/pythonstackProject/venv/bin/python /Users/cheonjiu/PycharmProjects/pythonstackProject/checkBrackets.py
입력 : (3+4)*(2+(30-7)+2133)
Wrong _ 5

Process finished with exit code 0
```

```python
class Stack_:
    def __init__(self):
        self.top = []

    def isEmpty(self): return len(self.top) == 0
    def size(self): return len(self.top)
    def clear(self): self.top = []

    def push(self, item):
        self.top.append(item)

    def pop(self):
        if not self.isEmpty():
            return self.top.pop(-1)

    def peek(self):
        if not self.isEmpty():
            return self.top[-1]

def checkBrackets(statement):
```

Run: checkBrackets

```
/Users/cheonjiu/PycharmProjects/pythonstackProject/venv/bin/python /Users/cheonjiu/PycharmProjects/pythonstackProject/checkBrackets.py
입력 : 1*{4*(2-792)/1} + [3*{4-2* (100 -7)}]
OK _ 10

Process finished with exit code 0
```

```python
class Stack_:
    def __init__(self):
        self.top = []

    def isEmpty(self): return len(self.top) == 0
    def size(self): return len(self.top)
    def clear(self): self.top = []

    def push(self, item):
        self.top.append(item)

    def pop(self):
        if not self.isEmpty():
            return self.top.pop(-1)

    def peek(self):
        if not self.isEmpty():
            return self.top[-1]

def checkBrackets(statement):
```

```
/Users/cheonjiu/PycharmProjects/pythonstackProject/venv/bin/python /Users/cheonjiu/PycharmProjects/pythonstackProject/checkBrackets.py
입력 : (01*{4+{2101-7}/1} + 9+{4-2+ (10188-7}}}
Wrong _ 9

Process finished with exit code 0
```

```python
class Stack_:
    def __init__(self):
        self.top = []

    def isEmpty(self): return len(self.top) == 0
    def size(self): return len(self.top)
    def clear(self): self.top = []

    def push(self, item):
        self.top.append(item)

    def pop(self):
        if not self.isEmpty():
            return self.top.pop(-1)

    def peek(self):
        if not self.isEmpty():
            return self.top[-1]

def checkBrackets(statement):
```

Stack > peek()

Run: checkBrackets

```
/Users/cheonjiu/PycharmProjects/pythonstackProject/venv/bin/python /Users/cheonjiu/PycharmProjects/pythonstackProject/checkBrackets.py
입력 : {3*(4001+(2·7)/3} + (3+(4·2+ (1·7)}])
OK _ 12

Process finished with exit code 0
```