괄호 회전하기

올바른 괄호 문자열 : () {} []

$s = \overset{1}{[]}\overset{2}{()}\overset{3}{\{\}}$ stack 1, 2, 3 로 각각 올바른지 체크

S는 왼쪽으로 x칸 만큼 이동 (0 ≤ x < s.length)

s.length = 6 ⇒ 0 ≤ x < 6

x=0일때,	x=1일때,	x=2일때	x=3일때	x=4일때	x=5일때
[](){}	](){}[	(){}[]	){}[](	{}[]()	}[](){
stack1, 2, 3 다 빔.	첫 요소가 닫는괄호	stack1, 2, 3 다 빔	첫 요소가 닫는괄호	stack1, 2, 3 다 빔	첫 요소가 닫는괄호
count=1		count =2		count = 3	

조기반환

s가 홀수이면 싹다 올바르지 않음. ⇒ 0 반환

s의 처음이 닫는괄호, s의 마지막이 여는 괄호면 넘어감.

$\begin{array}{l} 4\ 5\ 0\ 1\ 2\ 3 \\ 5\ 0\ 1\ 2\ 3\ 4 \\ 0\ 1\ 2\ 3\ 4\ 5 \end{array}$

s 회전 어떻게? [](){}

[: 0 → 5 → 4 → 3 → 2 → 1

i=0, x=2, len=6

+5 ⇒ 5, 6, 7, 8, 9, 10

$6\overline{)5}$ 0…5 , $6\overline{)6}$ 1…0 $6\overline{)7}$ 1…1 $6\overline{)10}$ 1…4

⇒ x×5를 더해.

1. s의 길이를 변수 len에 담고 홀수인지 확인 (홀수이면 올바른 괄호 나올수 없음)

```
int len = s.length();
if (len % 2 != 0){
   return 0;
}
```

2. 올바른 괄호 찾기 위해 Stack 생성

```
Stack<Character> smallParen = new Stack<>();
Stack<Character> curlyBrace = new Stack<>();
Stack<Character> squareBracket = new Stack<>();
```

3. x번 회전시킨 s를 담기 위한 CharArray, 올바른 괄호 문자열의 개수 담을 count 변수 생성

```
Char[] charArr = new Char[len];
int count =0;
```

4. x번 회전시켜야 하므로 for문 이용

```
for (int x =0; x < len; x++){
```

5. x번 회전시킨 s를 charArr에 담기

```
    for(int i=0; i< len; i++){
        charArr[(i+(x×5))%len] = s.charAt(i);
    }
```

6. 조기 반환 : 0번 인덱스) or } or] 혹은 len-1번 인덱스 (or { or [⇒ 다음 x로 이동

```
    if(charArr[0]==')' || charArr[0] == '}' || charArr[0] == ']' || charArr[len-1] == '(' || charArr[len-1] == '{' || charArr[len-1] == '['){
        continue;
    }
```

7. 올바른 괄호 로직

```
for(int i=0; i<len; i++){
    char ch = charArr[i];
```

8. ch가 (, {, [이면 스택에 추가

```
    if(ch == '('){
        smallParen.push(ch);
    } else if(ch == '{'){
        curlyBrace.push(ch);
    } else if(ch == '['){
        squareBracket.push(ch);
    }
```

9. 해당하는 괄호 스택이 비어있지 않고 ')', '}', ']'이면 pop, 비어있으면 다음 진행.

```
    if(ch == '('){
        if(!smallParen.isEmpty()){
            smallParen.pop();
        } else{
            continue;
        }
    }else if(ch == '{'){
        if(!curlyBrace.isEmpty()){
            curlyBrace.pop();
        }else{
            continue;
    } else if(ch == '['){
        if(!squareBracket.isEmpty()){
            squareBracket.pop();
            } else {
                continue;
            }
        }
```

10. 세 개의 스택이 다 비어있으면 올바른 괄호 문자열 이므로 count++

```
    if (smallParen.isEmpty() && curlyBrace.isEmpty() && squareBracket.isEmpty()){
        count++;
    }
  }
}

return count;
```