

# 소프트웨어 입문 설계

## 실습 문제 34

### Calculator

+, -, 괄호를 지원하는 계산기를 구현하시오. +, - 는 바이너리 연산자만 지원하면 됨.  
즉 unary 연산자는 없다고 가정하시오. 아래와 같이 계산 과정을 반드시 출력하시오.  
뒷장의 설명을 참고하세요.

입력)  
공백문자(space)를 포함하지 않는 수식

실행 예제 1 - 괄호 없는 경우)

```
Enter equation:
4-3+4+1233213-23
STARTING:4-3+4+1233213-23
+4=4
-3=1
+4=5
+1233213=1233218
-23=1233195
FINISHING:4-3+4+1233213-23
The answer is: 1233195
```

실행 예제 2 - 괄호 있는 경우

```
Enter equation:
(234-123)+1323-(123+43)
STARTING:(234-123)+1323-(123+43)
STARTING:234-123
+234=234
-123=111
FINISHING:234-123
+111=111
+1323=1434
STARTING:123+43
+123=123
+43=166
FINISHING:123+43
-166=1268
FINISHING:(234-123)+1323-(123+43)
The answer is: 1268
```

실행예제 3 - 중첩된 괄호 있는 경우

```
Enter equation:
(34 - (234+33)+34)+47
STARTING: (34 - (234+33)+34)+47
STARTING:34 - (234+33)+34
+34=34
STARTING:234+33
+234=234
+33=267
FINISHING:234+33
-267=-233
+34=-199
FINISHING:34 - (234+33)+34
+-199=-199
+47=-152
FINISHING: (34 - (234+33)+34)+47
The answer is: -152
```

구현 방법: recursion 을 사용하세요.

1. 실행 예제 1 이 동작하도록 먼저 계산기 함수를 구현

예를 들어 `int calc(const char* string, int startIndex, int endIndex);` 함수를 구현한다.

이 함수는 수식이 저장된 문자열 배열의 시작 index 와 끝 index 사이에 해당하는 부분 수식을 계산한다. 위 출력에서 STARTING:문장은 calc 함수 호출 시작부분에서, FINISHING: 문장은 함수 종료 직전에 출력된 것이다.

2. 괄호가 나오는 경우 괄호 안을 같은 함수를 사용하여 먼저 계산하는 방식으로 확장 한다. 중첩된 괄호의 경우 짝이 맞는 괄호를 찾아서 같은 함수 사용.

예를 들어  $(34-(234+33)+34)+47$  의 경우 첫 ( 가 발견되면 짝이 맞는 괄호, 즉 34 뒤의 두번째 닫는 괄호 ) 를 찾을 수 있고, (와 )사이를 계산기 함수 calc 에 넣어 계산한다. recursion 에 의해  $(234+33)$ 부분도 같은 함수를 사용하여 계산한다.