

과제1: 수식계산 함수 작성하기

아래 첨부이미지와 같은 소스코드가 있다. eval함수는 main함수로부터 수식 문자열을 받아서 수식 결과를 계산하여 돌려주는 함수이다. main함수에서 exp 변수값에 테스트하고 싶은 수식 문자열을 넣어 eval함수를 테스트할 수 있다. 다음은 eval함수 작성과 관련된 요구조건 및 가정사항이다. 아래 조건들을 고려하여 eval함수를 작성하시오.

- main함수에서 eval 함수를 실행시켜 수식 계산 결과를 출력할 수 있는 eval 함수를 완성하도록 한다(코드샘플 첨부 이미지 참고)
- 인자로 전달받은 문자열 수식 속에는 피연산자, 사칙연산자(+ - * /), 괄호 기호만 등장한다. 괄호 기호는 소괄호(둥근괄호)만 사용한다고 가정한다.
- 문자열 속 피연산자는 1~99 사이의 정수라고 가정한다. 즉, 피연산자는 한자리 숫자일 수도 있고 두 자리 숫자일 수도 있다.
- 입력으로 주어지는 수식 문자열은 괄호 에러를 가지고 있을 수 있다. 예를 들어 "2*(3+2))" 또는 "2*(3-1"와 같이 괄호 짝이 맞지 않을 수 있다. 따라서 수식 계산에 앞서 괄호검사를 수행하고 만약 괄호 에러가 발생한 수식이 입력되었다면 계산을 수행하지 않고 -999 값을 리턴한다. 그러나 괄호 에러가 없다면 주어진 수식을 계산하여 계산 결과를 리턴한다.
- 수식의 계산 결과는 -999가 되는 경우는 없다고 가정한다. 즉, 계산 결과가 -999가 되는 수식은 테스트케이스로 입력되지 않는다고 본다. 여기서 -999는 에러상황을 표현하기 위한 따로 약속된 정수값이다.
- 만약 수식 계산 도중 어떤 숫자를 0으로 나눠야 하는 경우가 발생하면 이 경우는 계산이 불가하므로 계산 과정을 중단하고 -999를 리턴한다. 예를 들어 "5/(10-10)"과 같은 수식은 계산 불가하다.
- 수식의 최종 결과는 경우에 따라 실수값이 될 수도 있다. 하지만 최종적으로 리턴되는 값은 소수점 이하를 버리고 정수값 형태로 전달한다. 즉, 예를 들어 수식 계산 결과가 3.14라면 3을 리턴하도록 한다.
- eval 함수의 프로토타입은 변경하지 않고 int eval(char *s)를 유지하고 함수 내용만을 작성하도록 한다.
- eval함수 안에서 별도의 외부 사용자 정의함수를 호출하거나 전역변수 또는 데이터타입 등을 참조하는 것은 가능하다. 즉, eval함수와 main함수 이외에 추가적으로 필요한 함수나 데이터타입이 있다면 eval함수 밖에서 정의하여 사용해도 된다. 단, 아래 첨부 이미지처럼 main함수에서 eval함수만을 단독으로 호출했을 때 수식계산이 완전하게 수행되어

결과를 넘겨 받을 수 있도록 eval함수를 작성한다.

- stdio.h stdlib.h string.h 이외의 외부 라이브러리는 사용할 수 없다. 예를 들어 수식계산과 관련된 외부라이브러리 함수의 힘을 빌려서 수식 계산을 할 수 없으며, 수업 때 배운 스택 자료구조와 알고리즘을 이용하여 직접 eval함수를 구현하도록 한다.

<코드샘플 첨부이미지>

```
1  #include <stdio.h>
2
3
4  int eval(char *s) {
5      // eval함수를 작성 하시오
6
7
8  }
9
10 int main(void) {
11     char *exp = "(12+3)*5";
12     int result;
13
14     result = eval(exp);
15     if (result == -999) {
16         printf("Error\n");
17         return 1;
18     } else {
19         printf("Result = %d\n", result);
20         return 0;
21     }
22 }
23
```