

## Week 3

- 주의 사항: 부정행위 금지, STL 사용 금지(string 사용가능), 배열, 연결 리스트를 이용하여 구현할 것.
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

### 문제 2

일반적으로 우리가 수식을 표기하는 방법은 다음과 같이 표기한다.

$$A + (B + C)$$

$$2 + (3 + 5)$$

이와 같이 피연자 사이의 연산자가 위치하는 표기법을 중위 표기법이라 한다. 컴퓨터에서 중위 표기 수식을 순서대로 계산할 경우, 연산자의 우선 순위를 고려하지 못하는 문제점이 생긴다. 예를 들면,  $2 + (3 + 5)$ 의 경우, 소괄호안에 있는  $+$ 연산을 우선적으로 처리 해야 하는데, 순서대로 계산 하게 되면, 소괄호를 먼저 처리할 수 없게 된다. 또한,  $2 * 3 + 5$ 의 경우,  $+$ 보다  $*$ 가 우선 순위가 높기 때문에 먼저 처리해야 하는데, 순서대로 계산하게 되면 잘못된 해를 구하게 된다. 이 때문에 컴퓨터 프로그램에서는 수식 계산을 쉽게 하기 위해 중위 표기법으로 표기된 수식을 다음과 같이 변환하여 사용한다.

$$A + (B + C)$$

$$3\ 5\ 2++$$

이처럼 피연산자를 먼저 표기하고 피연산자 뒤에 연산자를 표기하는 방법을 후위 표기법이라 한다.

홍준은 중위 표기를 입력했을 때, 후위 표기로 변환 해주는 프로그램을 만들려고 한다.

### 입력

표준 입력으로 다음과 같이 주어진다.

첫째 줄에 주어지는 수식의 수  $N$  ( $1 \leq N \leq 10,000$ )이 주어진다. 둘째 줄부터  $N$ 개의 줄에는 각각 중위표기법으로 적힌 수식이 띄어쓰기 없이 주어진다. 이 때 수식은 정수( $1 \leq N \leq 9$ )와 연산자( $+$ ,  $-$ ,  $*$ )로만 이루어진다. (단 주어지는 수식의 길이  $P$  ( $3 \leq P \leq 10000$ ))

### 출력

출력해야 하는 중위 연산식이 주어질 때마다 그 결과를 한 줄에 하나씩 출력한다.

## 예제 입출력

예제 입력	예제 출력
5	12+5*
(1+2)*5	35*2+
3*5+2	167-*32-+
1*(6-7)+(3-2)	284*5*-
2-8*4*5	15-76*-
1-5-(7*6)	

## HINT

모든 중위 표기식을 스택에 넣은 후 조건에 맞춰서 pop하는 방식이 아닌, 중위 표기식으로 입력된 값을 순서대로 읽으면서 조건에 맞춰 스택에 연산자와 소괄호를 push or pop하여 후위 표기식으로 변환하는 방식이다. 중위 표기식을 읽을 때, 입력된 값이 연산자인지 정수인지 구별하는 함수와 연산자의 우선순위 및 비교하는 함수는 다음과 같다.

```
bool isOper(char item) {
    if (item == '+' || item == '-' || item == '*') {
        return true;
    }
    else return false;
}

bool isNum(char item) {
    int i;
    if (typeid(item - '0') == typeid(int)) {
        return true;
    }
    else return false;
}

int getOper_order(char op) // 연산자의 우선순위 반환하는 함수
{
    switch (op)
    {
        case '*':
            return 2;
        case '+':
            return 1;
        case '-':
            return 1;
    }
}

int compareOper(char op1, char op2) // 우선순위 비교함수
{
    int op1Prec = getOper_order(op1);
    int op2Prec = getOper_order(op2);

    if (op1Prec > op2Prec)
        return 1; // op1의 우선순위가 더 높다면
    else if (op1Prec < op2Prec)
        return -1; // op2의 우선순위가 더 높다면
    else
        return 0; // 우선순위가 같다면
}
```