

## C. 후위표기법

(Time Limit: 10 seconds)

우리가 일반적으로 사용하는 계산방식은 중위표기법(Infix Notation)이라 하는데,  $A + B$ 와 같이 피연산자 'A'와 'B' 중간에 연산자 '+'가 위치하여 이렇게 불린다. 컴퓨터공학에서는 후위표기법(Postfix Notation)을 많이 사용하는데, 후위표기법은  $A B +$  와 같이 피연산자 'A'와 'B'의 뒤에 연산자 '+'가 위치한 표기법을 말한다.

중위표기법에서  $(5 + 8) * 2$  와 같은 수식은 '\*'가 '+'보다 연산자 우선순위가 높으므로 앞의 수식에서처럼  $5 + 8$  을 먼저 계산해야 한다면 괄호를 사용해야 한다. 하지만 수식  $(5 + 8) * 2$  을 후위표기법으로 바꾸면  $5 8 + 2 *$  와 같이 되어, 후위표기법은 괄호가 없이도 연산자의 우선순위를 명확히 할 수 있다는 장점이 있다. 이런 이유로 소프트웨어로 구현되는 계산기들은 후위표기법을 많이 사용한다. 그럼 후위표기법의 수식을 입력 받아 계산하는 프로그램을 작성해 보자. 또한, 계산기는 예외처리에도 신경을 써야 하므로 만약 0으로 나누는 경우가 발생하면 계산을 중단한다.

### Input & Output

첫 번째 줄에는 테스트케이스의 개수  $T$  ( $0 < T \leq 50$ )가 입력된다.

각 테스트 케이스는 두 줄으로 구성되는데, 첫 줄은 총 입력되는 연산자와 피연산자의 개수의 합  $M$  ( $3 \leq M \leq 11$ )이 입력되며, 둘째 줄은  $M$ 개 만큼의 연산자와 피연산자가 한 칸씩의 공백을 두고 입력된다. 피연산자  $X$  는  $-30 \leq X \leq 30$  의 범위를 가지는 정수이며, 연산자는 사칙연산인 '\*', '/', '+', '-'의 네 가지가 입력되고, 계산 결과 값 및 계산 중간 과정에서의 값은 integer 범위 ( $-2,147,483,648 \sim 2,147,483,647$ ) 내의 값이 된다. 만약 0으로 나누는 경우가 생기면 "NO"를 출력한다.

### Notes

피연산자나 연산자가 부족한 경우와 같이 완전하지 않은 수식은 입력되지 않는다.

나눗셈 연산의 경우, 소수점 이하는 버리고 몫만 계산되며 정수로 나타난다. 예를 들어  $10 / 3$  은 3 이 된다.

Sample Input	Output for Sample Input
5	5
3	26
2 3 +	NO
5	0
5 8 + 2 *	3
3	
3 0 /	
3	
0 3 /	
3	
10 3 /	