



# 호출 순서

**[문제]** 몇 개의 함수로 구성된 프로그램이 있다. 이 프로그램은 `main( )`에서 시작하여 수행된다. 이 수행 도중 다른 함수 호출(function call)이 다층적으로 포함되어 있다. 우리는 입력으로 제시된 코드에서 각 문장(statements)이 실행(execution)되는 순서를 추적하고자 한다.

`main( )`을 포함한 모든 함수는 함수 이름과 그 문장의 내용으로 구성되어 있다. 함수 내용은 그 안에서 호출(call)되는 다른 함수를 포함한 수행문(statements)의 순서(sequence)로 표현되어 있다. 모든 함수 이름은 **영어 대문자 1글자**로 표시되며, 수행문은 소문자 문자열(string)로 표시되어 있다. 만일 어떤 함수의 수행 중에 다른 함수를 호출할 경우 이 호출은 해당 함수의 이름, 즉 영문 대문자 한 글자로 표시된다. 그리고 각 함수의 마지막(end)은 특수 문자 '\$'로 표시되어 있으며 이 문자를 만나면 해당 함수는 `return` 종료된다. 아래는 4개의 함수 **M, G, K, W**로 구성된 프로그램의 예이다.

<b>M( )</b>	i k pop push <b>G</b> bab ko mong <b>K</b> viva <b>\$</b>
<b>G( )</b>	so good bad <b>K</b> baby you <b>W</b> mart <b>\$</b>
<b>K( )</b>	killling me <b>W</b> <b>W</b> nice day <b>\$</b>
<b>W( )</b>	data structure kills every body <b>\$</b>

시작은 항상 **M(Main)** 함수로부터 시작한다. 각 함수에서의 단위 작업(basic execution)은 함수 이름 (F)에 '-'(minus) 기호로 연결하여 '**F-act**'와 같이 표현해야 한다. 만일 위에서 제시된 프로그램이라면 각 문장의 수행 순서는 다음과 같다.

M-i	M-k	M-pop	M-push	G-so	G-good
G-bad	K-killling	K-me	W-data	W-structure	
W-kills	W-every	W-body	W-data	(...)	
W-body	K-nice	K-day	G-baby	G-you	W-data
(...) W-body	G-mart		M-bab	M-ko	M-mong
K-killling	K-me		W-data (...중략...)		
W-body	W-data (...)	K-nice	K-day	M-viva	

여러분에게는 전체 프로그램과 함께 어떤 순서를 나타내는 두 개의 정수  $k_1$   $k_2$ 가 주어진다. 여

러분은 이 프로그램이 수행될 때  $k_1$   $k_2$  번째에 해당하는 수행문을 'F-act'와 같은 방식으로 하나의 문자열로 각각 출력해야 한다. 단  $k$ 가 음수인 경우에는 그 절대값은 수행 순서의 뒤에서부터의 순서이다. 즉  $k = -1$ 이면 제일 마지막 문장,  $k = -2$ 이면 뒤에서 2번째 문장을 말한다.

만일 해당 순서에 대응하는 수행문이 없을 경우에는 문자열 '**NONE**'을 출력한다. 예를 들어 전체 나열된 수행문의 수가 100이고  $k_1$   $k_2$  이 각각 150, 200이라면 여기에 해당되는 수행문은 없으므로 '**NONE**'을 출력해야 한다. 프로그램이 유한 횟수 안에 종료되지 않고 무한루프(infinite loop)가 발생하면 문자열 '**DEADLOCK**' 출력해야 한다. 예를 들어 recursive call이 발생하면 무한루프가 발생하여 수행문의 수가 무한히 커지게 된다.<sup>1)</sup> 아래 예를 참조하시오.

**[입출력]** 입력과 출력은 모두 표준 입출력을 사용한다. 입력 파일 **stdion** 첫 줄에 3개의 정수가 주어진다. 3개 정수는 **main()**을 포함한 함수의 갯수  $N$ , 그리고 두 정수  $k_1$   $k_2$ 이다. 단  $2 \leq N \leq 20$ 이다.

이어지는  $N$ 개의 줄에는 각 함수의 내용이 제시된다. 각 줄의 첫 항목은 함수의 이름(대문자 1글자)이며 이어서 개별 수행 문장이 순서대로 한 줄에 모두 제시된다. 개별 수행 문장은 모두 소문자와 숫자로 구성된 문자열이다. 함수 당 수행 문장의 수는 2개 이상 최대 30개이다. 함수 중간에 call된 함수가 그 이름으로 표시되어 있으며 call된 함수는 항상 정의되어있다고 가정한다.

stdin	stdout
4 4 7 // N, k1 k2 M b W k S d w \$ S ee ww G ss \$ W k G q a \$ G 88 99 \$	G-99 M-k // 7번째
4 4 -3 // N, k1 k2=-3 M b W k S d w \$ S ee ww G ss \$ W k G q a \$ G 88 99 \$	G-99 S-ss // 뒤에서 3번째
4 8 70 // N, k1 k2 M b W k S d w \$ S ee ww G ss \$ W k G q a \$ G 88 99 \$	S-ee NONE
4 2 6 // N, k1 k2 M b W k S d w \$ S ee W ww G ss \$ W k G q a \$ G 88 99 S \$	DEADLOCK

**[제한조건]** 프로그램의 이름은 **callseq.{c, cpp, py}**이며 제출횟수는 15회, 수행시간은 최대 1초이다. C++ 사용자는 STL stack를, python 사용자는 abstract data structure로 STACK을 만들어 사용하기를 권한다.

1) 이 문제에서 recursive function의 종결 조건은 따로 지정되지 않는다고 가정한다.