

## 4. 탑 쌓기

윗면에 구멍이 파인 블록  $n$ 개가 주어진다. 블록  $i$ 의 높이는  $h_i$ , 구멍의 깊이는  $d_i (< h_i, h_{i+1})$ 이다. ( $h_i \geq 1, d_i \geq 0, h_i, d_i$ 는 정수) 블록  $i+1$ 의 아랫면은 정확히 블록  $i$ 의 윗면의 구멍과 맞물리지만, 그 이외 블록의 아랫면은 블록  $i$ 의 구멍에 조금이라도 들어갈 수 없다. 따라서 블록  $i$  위에 다른 블록을 올려서 이층 탑을 만들 때, 블록  $i+1$ 을 올리면 높이는  $h_i - d_i + h_{i+1}$ 이 되고 블록  $j (j \neq i, i+1)$ 를 올리면 높이는  $h_i + h_j$ 가 된다. 그림 1은 연속한 세 블록의 모양의 예시를 단면도로 보인다.

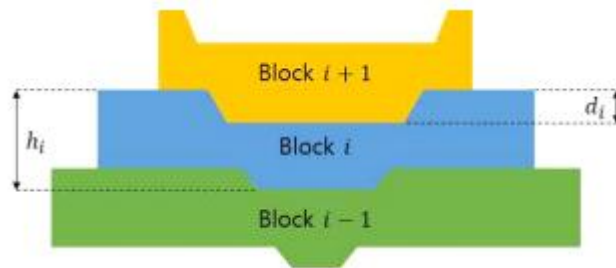


그림 1

이제 다음 조건들을 만족시키면서 블록들 중 일부를 선택하여 탑을 쌓으려 한다.

- 1)  $i < j$ 면 블록  $i$ 는 무조건 블록  $j$ 보다 탑의 아래층에 있어야 한다.
- 2) 탑의 높이는 주어진 자연수  $H$  이하여야 한다. ( $H \geq 1$ )

물론 탑의 각 층에는 한 개의 블록만 있고, 탑을 쌓을 때 블록을 뒤집거나 비스듬히 세울 수 없다. 만약 블록 1, 블록 2, 블록 5, 블록 6, 블록 7, 블록 9를 선택하여 탑을 만들었다면 그 높이는  $h_1 - d_1 + h_2 + h_5 - d_5 + h_6 - d_6 + h_7 + h_9$ 가 된다.

주어진  $h_i, d_i, H$ 에 대해, 위 조건대로 탑을 쌓을 수 있는 모든 경우의 수(블록 조합의 수)를 구하는 알고리즘을 작성하라, 단, 경우의 수가 int 최대값  $2^{31} - 1$ 를 초과할 가능성이 있으므로, 대신에 해당 값을 1,000,000으로 나눈 나머지를 구하는 것으로 한다.

### [제약사항]

- 시간 제한: 10개의 입력 전부를 해결하는 시간 총 1초
- 메모리 제한: 128MB
- 블록의 개수  $n$ 은  $[10, 1000]$  범위의 정수, 높이  $H$ 는  $[100, 10000]$  범위의 정수
- 각 블록의 높이  $h_i$ 는  $[1, 100]$  범위의 정수 ( $0 \leq i \leq n-1$ )
- 각 블록의 구멍의 깊이  $d_i$ 는  $[0, \min(h_i, h_{i+1}) - 1]$  범위의 정수 ( $0 \leq i \leq n-2$ )

### [입력]

입력 파일에는 10개의 테스트 케이스가 주어진다. 각 케이스는 세 줄로 이루어진다. 첫 줄에는 블록의 개수  $n$ 과 높이  $H$ 가 주어지고, 둘째 줄에는  $h_i$  값들이 주어진다. 셋째 줄에는  $d_i$  값들이 주어진다. 각각의 값은 공백으로 구분된다. 입력 파일의 이름은 “input.txt”이다.

[출력]

각 테스트 케이스에 대해서, 케이스의 번호를 “#x” 의 형식으로 출력한 후(여기서 x는 테스트 케이스 번호), 공백을 하나 둔 다음 주어진 케이스에서 답을 찾을 수 있는 모든 경우의 수를 1,000,000으로 나눈 나머지를 출력한다. 출력 결과물은 “output.txt”로 저장한다.

[예제]

입력 (input.txt)

10 100	← First Case
49 64 69 76 45 40 50 36 57 87	
18 13 58 30 30 31 26 32 50	
20 200	← Second Case
75 87 34 5 24 16 7 91 50 13 15 66 94 93 37 63 19 80 12 62	
65 17 4 3 12 1 0 26 3 0 4 4 69 3 3 1 11 9 1	
...	

출력 (output.txt)

#1 34
#2 20825
...