JAVA 03. WORKSHOP



✓ 배열과 정렬 알고리즘을 통한 응용력 향상

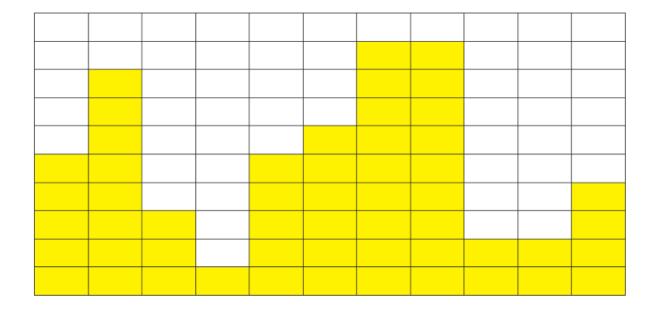
1

한 쪽 벽면에 다음과 같이 노란색 상자들이 쌓여 있다.

높은 곳의 상자를 낮은 곳에 옮기는 방식으로 최고점과 최저점의 간격을 줄이는 작업을 평탄화라고 한다.

평탄화를 모두 수행하고 나면, 가장 높은 곳과 가장 낮은 곳의 차이가 최대 1 이내가 된다.

평탄화 작업을 위해서 상자를 옮기는 작업 횟수에 제한이 걸려있을 때, 제한된 횟수만큼 옮기는 작업을 한 후 최고점과 최저점의 차이를 반환하는 프로그램을 작성하시오.

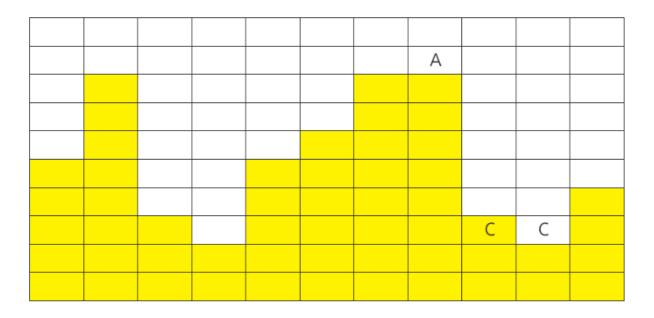


가장 높은 곳에 있는 상자를 가장 낮은 곳으로 옮기는 작업을 덤프라고 정의한다. 위의 예시에서 제1회 덤프를 수행한 이후 화면은 다음과 같다.

			Α	А		
	В					

JAVA 02. WORKSHOP

A부분의 상자를 가장 낮은 B부분에 덤프하였으며, A대신 A'부분의 상자를 사용해도 무방하다. 다음은 제2회 덤프를 수행한 이후의 화면이다.



A'부분의 상자를 옮겨서, C부분에 덤프하였다. 이때 C 대신 C'부분에 덤프해도 무방하다.

2회의 덤프 후, 최고점과 최저점의 차이는 8 - 2 = 6 이 되었다 (최초덤프 이전에는 9 - 1 = 8 이었다).

덤프 횟수가 2회로 제한된다면, 이 예시 문제의 정답은 6이 된다.

[제약 사항]

가로 길이는 항상 100으로 주어진다.

모든 위치에서 상자의 높이는 1이상 100이하로 주어진다.

덤프 횟수는 1이상 1000이하로 주어진다.

주어진 덤프 횟수 이내에 평탄화가 완료되면 더 이상 덤프를 수행할 수 없으므로 그 때의 최고점과 최저점의 높이 차를 반환한다 (주어진 data에 따라 0 또는 1이된다).

[입력]

총 10개의 테스트 케이스가 주어지며, 각 테스트 케이스의 첫 번째 줄에는 덤프 횟수가 주어진다. 그리고 다음 줄에 각 상자의 높이값이 주어진다.

[출력]

#부호와 함께 테스트 케이스의 번호를 출력하고, 공백 문자 후 테스트 케이스의 최고점과 최저점의 높이 차를 출력한다.

입력

834

42 68 35 1 70 25 79 59 63 65 6 46 82 28 62 92 96 43 28 37 92 5 3 54 93 83 22 17 19 96 ...

617

16 40 59 5 31 78 7 74 87 22 46 25 73 71 30 78 74 98 13 87 91 62 37 56 68 56 75 32 53 ...

...

출력

#1 13 #2 32

...