



야적장(Yard)¹⁾

[문제] 태풍이 예고된 부산항 야적장에 쌓인 컨테이너를 아래 규칙으로 옮겨 사고를 대비한다.

- (1) 컨테이너는 항상 가장 위(top)에 있는 것만 움직일 수 있다.
- (2) 가장 높은 위치의 컨테이너를 들어내어 가장 낮은 빈자리 자리로 옮긴다.
- (3) 같은 높이의 컨테이너가 하나 이상이면 더 무거운(큰 숫자)을 선호한다.²⁾
- (4) 가장 낮은 자리가 하나 이상이면 적치장 번호가 더 빠른(작은) 곳을 선호한다.³⁾
- (5) 적치장의 최고, 최저 높이의 차이가 1 이하가 되면 이 정리 작업은 멈춘다.

아래 그림을 보면서 설명을 해보자. Step-0은 야적장의 초기 상태를 보여준다. 각 박스는 하나의 컨테이너를 나타내며 그 안에 들어있는 숫자는 해당 컨테이너의 무게(tons)을 의미한다.

작업을 시작하면 먼저 가장 높은 위치에 있는 컨테이너를 찾아야 한다. 이 경우 3번째 적치장의 꼭대기, 즉 높이 5에 있는 컨테이너 C[19]를 들어내서 가장 낮은 장소로 옮겨야 한다. 이 경우 현재 가장 낮은 높이의 적치장은 2군데 {2, 6}이므로, 이렇게 같은 높이가 있을 경우에는 더 앞쪽에 우선 배치해야 하므로 우리는 C[19]를 적치장 2번으로 옮겨야 한다. 이제 전체 Shipping yard의 상태는 그림 Step-1에 있다.

Step-1에서 보면 가장 높은 곳에 배치된 컨테이너는 2개이다. 즉 1번, 3번 적치장에 있는 {81, 40}인데, 이 경우에는 규칙에 의하여 더 무거운 C[81]부터 먼저 이동시켜야 한다. 따라서 C[81]을 찾아서 가장 낮은 위치인 6번 적치장 바닥에 배치한다. 그 다음에는 3번 적치장 top의 C[40]을 가장 낮은 높이의 적치장 중에서 제일 앞 쪽, 즉 2번 적치장에 내려 놓는다. 그 결과는 Step-3과 같다. 그리고 Step-4에서 C[15]를 6번 적치장에 옮기고 나면, 어제 전체 Yard에서 가장 높은 적치장과 가장 낮은 적치장의 높이 차이가 1이 되므로 작업이 종료된다.

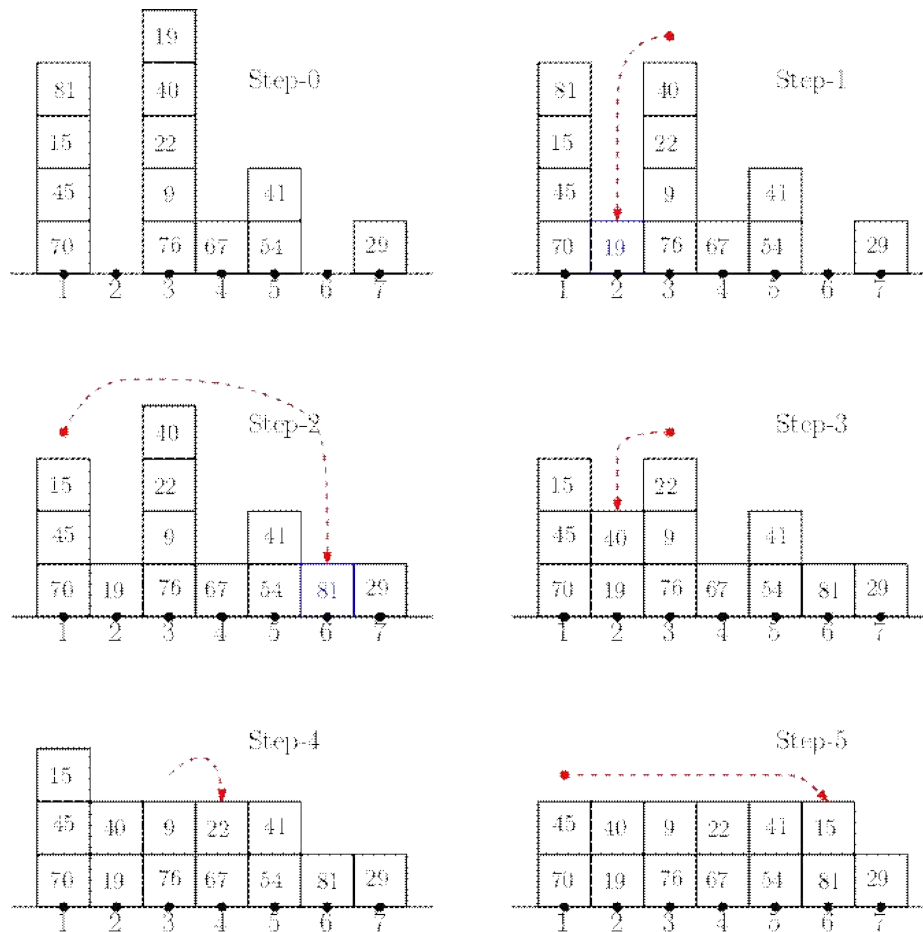
[입출력] 표준 입력(stdin) 첫 줄에 적치장의 개수 N ($2 \leq N \leq 50$)가 나오며 이어지는 N 개의 줄에는 i 번째 적치장에 쌓인 컨테이너 갯수 s_i 와 각 컨테이너의 ID(무게)가 밑에서부터 위로 s_i 개 주어진다. 출력은 1번 적치장부터 각 적치장에 쌓인 컨테이너의 ID(무게)를 바닥에서 위로 출력한다. 단 비어

1) 일반적으로 Shipping yard라고 부른다.

2) 위에 무거운 것이 있으면 더 위험하므로 이것부터 정리해야 더 안전해지기 때문이다.

3) 적치장은 선형으로 배열되어 있으며 1번부터 연속된 번호로 부여된다.

있을 경우에는 그 줄에 숫자 '0'을 출력해야 한다.



[예제]

입력 stdin	출력 stdout
7	70 45 //1번 적치장
4 70 45 15 81	19 40
0 // 비어있는 2번	76 9
5 76 9 22 40 19	67 22
1 67	54 41
2 54 41	81 15
0 // 비어있는 6번	29 //7번 적치장
1 29	
8	65
0 //빈 1번 적치장	45
0	31
0	65 10
4 65 10 30 45	65
1 65	30
0	15
0 //빈 7번 적치장	61 //8번 적치장
4 61 15 31 65	

[제한조건] 프로그램의 이름은 **yard.{py,c,cpp,java}**이며 python의 경우 numpy나 기타 package를 사용할 수 없다. 사용할 수 있는 토큰의 최대 개수는 **450개**이다.