

과제물 등수

[문제] 1부터 N 까지 순차 번호로 표시되는 수강생 N 명의 등수를 과제물 점수로 결정한다. 학생들은 최대 k 개까지의 과제물을 제출하였다. 각 과제물의 점수는 0점부터 10점까지 배정된다. 수강생의 등수는 다음과 같은 순서대로의 규칙으로 결정한다. 단 $1 \leq k \leq 30$ 이다.

규칙 1) 점수에 상관없이 더 많이 과제물을 낸 사람이 적게 낸 사람보다 앞선다.

규칙 2) 제출 과제물 수가 같다면 최저 점수가 높은 사람이 더 앞선다.

규칙 3) 만일 최저 점수가 같다면 **나머지에 대해서도 차례로 최저점수를 비교하여 결정한다.**

예) {3, 4, 2, 2}, {2, 2, 3, 5}일 때, {2, 2, 3}까지 같지만, 마지막이 4<5 이므로 두 번째 학생이 앞선다.

규칙 4) **모든 원소가 같다면** 번호가 빠른 사람이 등수에서 더 앞선다.

학생	제출 과제물 점수									
1	3	5	2	0	0	1	9			
2	10	10	10	10	2					
3	8	8	8	8	8					
4	2	3	6	7	8	1	2	9	2	
5	9	9	2							
6	6	9	9	9	8					
7	8	7	6	7	10					
8	6	6	6	6	6	6				

만일 위와 같다면 1등은 9개의 과제물을 낸 학생[4]이다. 그 다음은 7번 낸 학생[1], 다음 3등은 6번 낸 학생[8]이 된다. 그 다음 등위로 5번 낸 학생은 모두 4명으로 { [2], [3], [6], [7] }번인데, 각 학생의 최저 점수는 차례로 { 2, 8, 6, 6 } 이므로, 순위는 5번 낸 학생들의 순위는 3→{6, 7}→2 순서가 된다. 이 때 학생[6]과 학생[7]은 그 다음 최저 점수를 비교하면 각각 8과 7이므로, 학생[6]이 학생[7]에 앞선다. 꼴찌는 단 3개의 과제만 낸 학생[5]가 된다. 따라서 전체 순서는 4→1→8→3→6→7→2→5 가 된다.

[입출력] **rank.inp**의 첫 줄에는 학생의 수 N 이 주어진다. 단 $3 \leq N \leq 100$ 이다. 학생들의 과제물은 최대 30개 이다. 이어지는 N 개의 줄에는 과제물 점수가 주어지고 그 마지막은 숫자 -1로 표시되어 있다. 여러분은 1등부터 N 등까지의 학생 번호를 **rank.out**의 각 줄에 순서대로 출력해야 한다.

[예제]

rank.inp											rank.out
8	//	N=8 명									4
3	5	2	0	0	1	9	-1				1
10	10	10	10	2	-1						8
8	8	8	8	8	-1						3
2	3	6	7	8	1	2	9	2	-1		6
9	9	2	-1								7
6	9	9	9	8	-1						2
8	7	6	7	10	-1						5
6	6	6	6	6	6	-1					

[조건] 프로그램의 이름은 rank.{c, cpp, java} (**python 제외**), 제출횟수는 15회, 제한시간은 1초이다. 마감시간은 **10월 3(수) 저녁 10시**, 제출은 9월 28일(금)부터 가능하다.