Problem F

Many computer games have high score lists, where the best achieved scores are stored in non-ascending order. The rank of a score in such a list is normally the position in the sorted list. But if several scores are equal, their rank is the smallest position of such a score in the sorted list. For example, if the high score list looks like:

100 90 90 80

then the ranks would be

1 2 2 4

Given the number of possible entries in the high score list (int places), a list of scores (int[] scores) and a new score (int newscore), write a method getRank which returns the rank of the new score within the high score list. If the score is too low to get a position on the high score list, your method should return -1. Note that in a case where all places on the high score list are already filled, an old score will only be replaced if the new score is better.

- places is between 10 and 50, inclusive.
- The number of elements in scores is between 0 and places, inclusive.
- Each element of scores is between 0 and 2000000000, inclusive.
- scores is sorted in non-ascending order.
- newscore is between 0 and 2000000000, inclusive.

See examples for input/output format.

```
F.IN
1
3
100 90 80
```

```
90
10


F.OUT
2
```