Solution for "iCow" Bronze January 2008

by Nitin Kumar

We simply simulate the required operation, keeping track of the rating of each song. In fact, since we will need to choose a song 1000 times and each song will have a maximum rating of about 10,000, we even have enough time to distribute points one by one. First, let's read the input, using 1-indexing to comply with the problem statement.

```
read N, T
array R[N+1]
for i from 1 to N:
     read R_i
```

Next, we need to choose a song and distribute points T times, so we'll write a loop for that:

```
for i from 1 to T:
```

Every iteration of this loop, the first thing we need to do is to choose the best song and output it. We can do this by iterating through all the songs and choosing the one with the highest rating.

```
bestSong = 1               //we can choose this arbitrarily to start
for j from 1 to N:         //note we cannot use the same loop variable as before
     if R_j > R_bestSong:
          bestSong = j
output bestSong
```

And to finish up, we need to distribute all the rating points. As noted before, we can just do this one by one, making sure to skip ourselves. We need to keep distributing points while we have some left, keep track of the current song, and give them a rating point if they are not the one we just chose. Make sure to reset to the beginning when you have gone past the last song!

```
currentSong = 1
while R_bestSong > 0:
     if currentSong != bestSong:          //if they are not the one we just
chose
          R_bestSong -= 1            //give a rating point
          R_currentSong += 1
     currentSong += 1                      //go to the next song
     if currentSong == n+1:          //and loop around if necessary
          currentSong = 1
```

Putting it all together, we get the full solution:

```
read N, T
array R[N+1]
for i from 1 to N:
     read R_i
for i from 1 to T:
     bestSong = 1
     for j from 1 to N:
          if R_j > R_bestSong:
               bestSong = j
     output bestSong
     currentSong = 1
     while R_bestSong > 0:
          if currentSong != bestSong:
               R_bestSong -= 1
               R_currentSong += 1
          currentSong += 1
          if currentSong == n+1:
               currentSong = 1
```