**The ACM International Collegiate Programming Contest 1998–1999**
**North-western Europe Regional Contest**
**Sponsored by IBM**

# Problem B

# Plane Spotting

Craig is fond of planes. Making photographs of planes forms a major part of his daily life. Since he tries to stimulate his social life, and since it's quite a drive from his home to the airport, Craig tries to be very efficient by investigating what the optimal times are for his plane spotting.
Together with some friends he has collected statistics of the number of passing planes in consecutive periods of fifteen minutes (which for obvious reasons we shall call 'quarters'). In order to plan his trips as efficiently as possible, he is interested in the average number of planes over a certain time period. This way he will get the best return for the time invested. Furthermore, in order to plan his trips with his other activities, he wants to have a list of possible time periods to choose from. These time periods must be ordered such that the most preferable time period is at the top, followed by the next preferable time period, etc. etc. The following rules define which is the order between time periods:

1. A period has to consist of at least a certain number of quarters, since Craig will not drive three hours to be there for just one measly quarter.
2. A period $P_1$ is better than another period $P_2$ if:
   - the number of planes per quarter in $P_1$ is higher than in $P_2$;
   - the numbers are equal but $P_1$ is a longer period (more quarters);
   - the numbers are equal and they are equally long, but period $P_1$ ends earlier.

Now Craig is not a clever programmer, so he needs someone who will write the good stuff: that means you. So, given input consisting of the number of planes per quarter and the requested number of periods, you will calculate the requested list of optimal periods. If not enough time periods exist which meet requirement 1, you should give only the allowed time periods.

**Input**
The input starts with a line containing the number of runs N. Next follows two lines for each run. The first line contains three numbers: the number of quarters (1–300), the number of requested best periods (1–100) and the minimum number of quarters Craig wants to spend spotting planes (1–300). The second line contains one number per quarter, describing for each quarter the observed number of planes. The airport can handle a maximum of 200 planes per quarter.

**Output**
The output contains the following results for every run:
- A line containing the text "Result for run <N>:" where <N> is the index of the run.
- One line for every requested period: "<F>-<L>" where <F> is first quarter and <L> is the last quarter of the period. The numbering of quarters starts at 1. The output must be ordered such that the most preferable period is at the top.

## Sample input

```
3
10 5 5
1 5 0 2 1 4 2 5 0 2
10 3 5
10 3 1 4 2 6 3 0 8 0
5 5 5
1 2 3 4 5
```

## Output for the sample input

```
Result for run 1:
4-8
2-8
6-10
1-8
2-6
Result for run 2:
1-6
1-7
1-9
Result for run 3:
1-5
```