

## Museum Visitors

A museum records the entry time and exit time of visitors coming and going each day. This produces N pairs where the first value is the entry time and second value the exit time. Using this log, find the time range(s) when there were the maximum number of visitors inside the museum and how many of them were inside during that time range. The entry and exit times of the visitor are inclusive. For instance, if one visitor entered at 10:31 and another left at 10:31, there were 2 visitors in the museum at 10:31.

### Exercise

- 1) Implement an application that:
  - a) Reads the entering and leaving time of customers from the input file. The location of the input file must be given as a command line parameter to the application. The input file is a text file where each line contains entering and leaving time of one visitor, separated by comma. For example:  
10:15,14:20  
  
The times listed in the file may not be sorted by entry or exit time. To test the application, the existing input file called "visitingtimes.txt" can be used.
  - b) Based on the data read from the input file, find the time range(s) when there were the maximum number of visitors in the museum and how many of them there were.
  - c) Output the time range(s) and number of visitors found to standard output in the following format: <start of time range>-<end of time range>;<number of visitors>. For instance:  
10:31-10:45;19
- 2) Document your solution. The document should contain a description of your algorithm and your reasons for solving the problem in the way that you did, as well as any limitations foreseen. It could also contain your evaluation of other possible solutions and your reasons for rejecting them over your chosen approach.

The solution will be evaluated as though it was part of a production system and will be used to judge your technical capabilities based on your level of experience.

The solution will be judged on the general quality of the code and documentation. In particular it will be judged on whether it produces the correct answer, the algorithm efficiency, appropriate use of standard libraries/APIs, readability and maintainability. It will also be evaluated based on its scalability for possible future extensions such as times being supplied with a much higher precision or over a greater time range.

Unless instructed to implement the solution in a particular language, you are free to choose something suitable. If the solution is supplied in C++, please ensure that the code is ANSI compliant as in general, solutions are evaluated on a Linux platform.