

## Performance and Test Fall 2008

### Contest description

In relation to your project you have possibility to win great prizes for the best performing implementation of the route finder.

**Problem:** Your program should be able to read in a file with a set of locations (start, end) for which you should output a new file containing the corresponding lengths of the shortest paths.

**Input file:** The input file consists of N lines. Each line contains 6 comma separated fields indicating the start and end address:

```
<StreetNameFrom>, <NrFrom> ,<Zip/PostNrFrom>, <StreetNameTo>, <NrTo>, <Zip/PostNrTo>
<StreetNameFrom>, <NrFrom> ,<Zip/PostNrFrom>, <StreetNameTo>, <NrTo>, <Zip/PostNrTo>
....
<StreetNameFrom>, <NrFrom> ,<Zip/PostNrFrom>, <StreetNameTo>, <NrTo>, <Zip/PostNrTo>
```

Where

<StreetNameFrom>: the street name of the start location.  
<NrFrom>: the street number of the start location.  
<Zip/PostNrFrom>: the zip code (postnumber) of the start location.

And similarly for the <...To> variables. You may use the Java StringTokenizer for parsing the strings.

**Output file:** The output file should also contain N lines. Each line contains the length of the shortest path your program has found between the locations in the corresponding line in the input file. Each line contains the length of the shortest path your program has found between the locations in the corresponding line in the input file (<Distance> in km), the time to go from A to B (<EstimatedTime> in minutes) and the time it took your program to calculate the path, <CalulationTime> (in seconds after having loaded the graph). i.e. <Distance>, <Estimated Time>, <CalulationTime>.

Example files are provided on the homepage.

*Input.txt* is the expected format for the inputs to your program

*Output.txt* is the output format you should provide us.

**Evaluation:** Your program will be evaluated on both the found length, estimated time and the execution time.

**Submission:** You should hand in your program with a description on how to run it on an input file and the filename of the corresponding output. The preferred format for the program is a jar file. See the following example for help on creating jar files:

<http://www.itu.dk/courses/SPT/F2008/Episode04.1/AntExample.zip>

Good luck  
Kristian and Dan