Evaluation

Assignment 1 grading scheme is as follows.

A grade: A submission completing the requirements of the assignment which is well-structured and clearly written. Global variables are not used. route_manager runs without any problems; that is, all tests pass and therefore no extraneous output is produced. This is a good submission that passes all the tests and does not have any overall quality issues. Outstanding solutions get an A+ (90-100 marks). Solutions that are not considered outstanding by the evaluator will get an A (85-89 marks). A solution with minor issues will be given an A- (80-84 marks).

B grade: A submission completing the requirements of the assignment. route_manager runs without any problems; that is, all tests pass and therefore no extraneous output is produced. The program is clearly written. Although all the tests pass, the solution includes significant quality issues. Depending on the number of qualitative issues, the marker may give a B+ (77-79 marks), B (73-76 marks) or a B- (70-72 marks) grade.

A submission with any one of the following cannot get a grade higher than B:

- Submission compiles with warnings
- Submission has 1 or 2 large functions
- Program or file-scope variables are used

A submission with more than one of the following cannot be given a grade of higher than B-:

- Submission compiles with warnings
- Submission has 1 or 2 large functions.
- Program or file-scope variables

C grade: A submission completing most of the requirements of the assignment. route_manager runs with some problems. This is a submission that present a proper effort but fails some tests. Depending on the number of tests passed, which tests pass and a qualitative assessment, a grade of C (60-64 marks) or C+ (65-69 marks) is given.

D grade: A serious attempt at completing requirements for the assignment (50-59 marks). route_manager runs with quite a few problems. This is a submission that passes only a few of the trivial tests.

F grade: Either no submission given, or submission represents little work or none of the tests pass (0-49 marks). No submission, 0 marks**.** Submissions that do not compile, 0 marks. Submissions that do not run, 0 marks. Submissions that fail all tests and show a very poor to no effort (as evaluated by the marker) are given 0 marks. Submissions that fail all tests, but represent a sincere effort (as evaluated by the marker) may be given a few marks.

In general, straying from the assignment requirements will be penalized severely.

Additional Criteria for Qualitative Assessment

IMPORTANT: Only submissions that satisfy **ALL** the criteria described below will be considered for an A+ grade.

Documentation and commenting: the purpose of documentation and commenting is to write information so that anyone other than yourself (with knowledge of coding) can review your program and quickly understand how it works. In terms of marking, documentation is not a large mark, but it will be part of the overall quality assessment.

Functional decomposition: quality coding requires the good use of functions. Code that relies on few large functions to accomplish its goals is very poor-quality code. Typically, a good program has a main function that does some basic tasks and calls other functions, that do most of the work. A solution that passes all tests, but contains all code in one or two large functions will not be given a grade better than a B. You must not use program-scope or file-scope variables.

Proper naming conventions: You must use proper names for functions and variables. Using random or single character variables is considered improper coding and significantly reduces code readability.

Debugging / Comment artifacts: You must submit a clean file with no residual commented lines of code or unintended text.

Quality of solution: the marker will access the submission for logical and functional quality of the solution. Some examples that would result in a reduction of marks: solutions that read the input files several times, solutions which represent the data in inappropriate data structures, solutions which scale unreasonably with the size of the input.