Test Plan:

1. **Introduction**
   1. Test Plan Objectives
      1. **To test whether the program works as intended. The specifics on how the test are done are written in the following points.**
      2. **To check whether the right truck is allotted when location is equidistance from two or more truck.**
      3. **To check if the right truck is called to deliver the package. To elaborate, the truck should travel the shortest distance to deliver the package.**
2. **Scope**
   1. **Only the delivery of package will be tested. Therefore, how the truck returns will not be tested.**
3. **Test Strategy**

**We try to cover all the practical test scenarios that could be called upon when using this application. The following things are going to be tested:**  
3.1. General Test  
3.2. Equidistance Test  
3.3. Test where no diversions are needed  
3.4. Test where diversions are needed  
3.5. Invalid destination and size  
3.6. Negative Test  
3.7. Null Test  
3.8. Negative Test

1. **Environment Requirements**
   1. All the header files and source files provided should be complied and integrated successfully before testing the program. Usually, it will be done in IED such as VS Code.
2. **Execution Strategy**
   1. The tests will be conducted in the order which is given in Test Strategy (3). This is to ensure that all the critical aspects of the program’s functionality are verified.
   2. Prepare all the necessary data for inputs. This includes all the valid and invalid inputs that are necessary to test each scenario.
   3. **Test Reporting**
      1. After the tests are complete, they should be sent to the team lead.
      2. Along with that all the bugs should be reported to the developer in the team and the team lead, so that the bugs can be fixed.
      3. Finally, after all the bugs are fixed all the reports should be made available to the whole team for quality assurance.
   4. If any more bugs are found by the team, they should send the bug report to the team lead. After that the team lead will combine all the reports and share it to the developers.
3. **Test Schedule**
   1. **The testing should start at least 4 days prior to the date of submission. Furthermore, the test report should be made available to the team so that all the members have time to properly assess the work. This also ensures that the developers have enough time to fix any bugs found by the team.**
4. **Control Procedures**
   1. 6.1 Reviews will be done before and after the test and reports are created. This is to make sure that all the requirements are met.  
      6.2 Team meetings will be held to resolve any issue related to testing or bug fixing. Moreover, to prioritize the reported bugs that are to be fixed.  
      6.3 Change request will be made by the team lead to the developer as part of the bug fixing process.  
      6.4 All the bugs found by any team member will be reported to the team lead and will be discussed in the team meeting.
5. **Functions To Be Tested**

The test cases that are provided in section 3 will test the following functions:

**To summarize all the functions provided in the program file will be tested**

1. populateMap

2. getNumRows

3. getNumCols

4. printMap

5. getBlueRoute

6. getGreenRoute

7. getYellowRoute

8. addRoute

9. addPtToRoute

10. addPointToRoute

11. addPointToRouteIfNot

12. distance

13. shortestPath

14. getPossibleMoves

15. getClosestPoint

16. eqPt

1. **Deliverables**
   1. **Test Plan**
   2. **Test Report**
   3. **Bug Report**
2. **Documentation**

All the details related to the test plan and bug fixes will be documented.

1. **Approvals**

**Before the final submission and final meeting will be held to ensure that all the team members are satisfied with the quality of work generated.**