

This task is the implementation phase of the WGUPS Routing Program.

The Western Governors University Parcel Service (WGUPS) needs to determine an efficient route and delivery distribution for their daily local deliveries (DLD) because packages are not currently being consistently delivered by their promised deadline. The Salt Lake City DLD route has three trucks, two drivers, and an average of 40 packages to deliver each day. Each package has specific criteria and delivery requirements that are listed in the attached "WGUPS Package File."

Your task is to determine an algorithm, write code, and present a solution where all 40 packages will be delivered on time while meeting each package's requirements and keeping the combined total distance traveled under 140 miles for two of the trucks. The specific delivery locations are shown on the attached "Salt Lake City Downtown Map," and distances to each location are given in the attached "WGUPS Distance Table." The intent is to use the program for this specific location and also for many other cities in each state where WGU has a presence. As such, you will need to include detailed comments to make your code easy to follow and to justify the decisions you made while writing your scripts.

The supervisor should be able to see, at assigned points, the progress of each truck and its packages by any of the variables listed in the "WGUPS Package File," including what has been delivered and at what time the delivery occurred.

## ASSUMPTIONS

---

- Each truck can carry a maximum of 16 packages, and the ID number of each package is unique.
- The trucks travel at an average speed of 18 miles per hour and have an infinite amount of gas with no need to stop.
- There are no collisions.
- Three trucks and two drivers are available for deliveries. Each driver stays with the same truck as long as that truck is in service.
- Drivers leave the hub no earlier than 8:00 a.m., with the truck loaded, and can return to the hub for packages if needed.
- The delivery and loading times are instantaneous (i.e., no time passes while at a delivery or when moving packages to a truck at the hub). This time is factored into the calculation of the average speed of the trucks.
- There is up to one special note associated with a package.
- The delivery address for package #9, Third District Juvenile Court, is wrong and will be corrected at 10:20 a.m. WGUPS is aware that the address is incorrect and will be updated at 10:20 a.m. However, WGUPS does not know the correct address (410 S. State St., Salt Lake City, UT 84111) until 10:20 a.m.
- The distances provided in the "WGUPS Distance Table" are equal regardless of the direction traveled.
- The day ends when all 40 packages have been delivered.

## REQUIREMENTS

---

*Your submission must be your original work. No more than a combined total of 30% of the submission and no more than a 10% match to any one individual source can be directly quoted or closely paraphrased from sources, even if cited correctly. The similarity report that is provided when you submit your task can be used as a guide.*

*You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.*

*Tasks may **not** be submitted as cloud links, such as links to Google Docs, Google Slides, OneDrive, etc., unless specified in the task requirements. All other submissions must be file types that are uploaded and submitted as attachments (e.g., .docx, .pdf, .ppt).*

Note: Use only appropriate built-in data structures, except dictionaries. You must design, write, implement, and debug all code that you turn in for this assessment. Code downloaded from the internet or acquired from another student or any other source may not be submitted and will result in automatic failure of this assessment.

- A. Develop a hash table, without using any additional libraries or classes, that has an insertion function that takes the package ID as input and inserts each of the following data components into the hash table:
  - delivery address
  - delivery deadline
  - delivery city
  - delivery zip code
  - package weight
  - delivery status (i.e., at the hub, en route, or delivered), including the delivery time
- B. Develop a look-up function that takes the package ID as input and returns *each* of the following corresponding data components:
  - delivery address
  - delivery deadline
  - delivery city
  - delivery zip code
  - package weight
  - delivery status (i.e., at the hub, en route, or delivered), including the delivery time
- C. Write an original program that will deliver *all* packages and meet all requirements using the attached supporting documents “Salt Lake City Downtown Map,” “WGUPS Distance Table,” and “WGUPS Package File.”
  1. Create an identifying comment within the first line of a file named “main.py” that includes your student ID.
  2. Include comments in your code to explain both the process and the flow of the program.
- D. Provide an intuitive interface for the user to view the delivery status (including the delivery time) of any package at any time and the total mileage traveled by all trucks. (The delivery status should report the package as at the hub, en route, or delivered. Delivery status must include the time.)

1. Provide screenshots to show the status of *all* packages loaded onto *each* truck at a time between 8:35 a.m. and 9:25 a.m.
  2. Provide screenshots to show the status of *all* packages loaded onto *each* truck at a time between 9:35 a.m. and 10:25 a.m.
  3. Provide screenshots to show the status of *all* packages loaded onto *each* truck at a time between 12:03 p.m. and 1:12 p.m.
- E. Provide screenshots showing successful completion of the code that includes the total mileage traveled by *all* trucks.
- F. Justify the package delivery algorithm used in the solution as written in the original program by doing the following:
1. Describe **two or more** strengths of the algorithm used in the solution.
  2. Verify that the algorithm used in the solution meets *all* requirements in the scenario.
  3. Identify **two** other named algorithms that are different from the algorithm implemented in the solution and would meet *all* requirements in the scenario.
    - a. Describe how *both* algorithms identified in part F3 are different from the algorithm used in the solution.
- G. Describe what you would do differently, other than the two algorithms identified in part F3, if you did this project again, including details of the modifications that would be made.
- H. Verify that the data structure used in the solution meets *all* requirements in the scenario.
1. Identify **two** other data structures that could meet the same requirements in the scenario.
    - a. Describe how *each* data structure identified in H1 is different from the data structure used in the solution.
- I. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.
- J. Demonstrate professional communication in the content and presentation of your submission.