Purpose: The purpose of this assignment is to provide a chance for you to demonstrate your ability to solve a job-relevant problem. There is no ‘correct’ solution, instead, we are interested in how you approach the problem; it will provide a basis for discussion during a technical interview.

Note: This is a new version of the assessment and, as a result, the intended difficulty or time required may not be accurate. Do your best in the recommended time. If any of the questions are unclear or there is a problem with the data please let us know.

Scenario: A drone is mounted with an internal GPS system and a 1D spinning LIDAR (something like this http://www.slamtec.com/en/lidar/a1). Every time the drone moves to a new location, the LIDAR does 1 full scan (sweep) of its surroundings. Our goal is to use the LIDAR data to improve the drone’s navigation. For this assignment altitude and drone orientation can be ignored. You can also assume that the scan is so fast that you can treat the drone as stationary for each sweep. One flight's worth of data is provided consisting of N sweeps over x seconds.

Tasks:

For this assessment complete any 2 of the following Tasks. It is recommended that you pick 2 Tasks that assist each other. Any language may be used however Java, C#, C++ or Python are preferred. You may use prebuilt libraries for any of your data processing if you wish.

1. *Display*

   Create a program to provide an appropriate visualization of the drone’s path and the LIDAR data. Ideally, the display should be able to show 1 sweep (1 scan ID) of data in isolation as well as all the sweeps combined together. This can be on separate displays or on the same display (with individual sweeps shown by highlighting for example)

   Input: LIDARDPoints.csv and FlightPath.csv (provided or created from another Task)

   Output: On-screen display
2. *Simulation*

   Generate new **LIDARDPoints** data based on a new room layout and new plausible flight plan. This data is not provided so you will need to create the layout and flight plan yourself. This can either be done manually (ensure you include your data with your submission) or programmatically.

   Input: Mapping.csv and FlightPath.csv (created, you may also use a map that matches the sample data provided however you will first need to generate this file (from part 5 for example))

   Output: LIDARDPoints**.**csv
3. *Flight optimization*

   Based on the data provided, find a better flight path that will result in the shortest possible travel time but still goes through the existing rooms. (Assume the first sampled location is the start point and the last sampled location is the end point)

   Input: LIDARDPoints.csv and FlightPath.csv (provided or created)

   Output: FlightPath.csv
4. *Flight reroute*

   Based on the data provided, find an alternative route that will take you to the end point faster. You may go through different rooms.

   Input: LIDARDPoints.csv and FlightPath.csv (provided or created)

   Output: FlightPath.csv
5. *Mapping*

   Use the multiple data sweeps to map out the dimensions of the rooms.

   Input: LIDARDPoints.csv and FlightPath.csv (provided or created)

   Output: Mapping.csv

Format:

- **Program IO:** As input, your program should take a path to the CSV files you will use as an input. As output, if the result is a data outputting Task, should be a csv, otherwise, if the Task is a visualization Task, it should display as it runs.
- **File Format:** Each time the drone takes a sample of data it generates a unique corresponding scan ID. This ID is shared between files and you can link location and lidar data using it.
  - **FlightPath:** FlightPath data is provided (and should be written in if you generate it) as a CSV file. The first line has the scan ID and number of data line (always 1). The next line is the X,Y location of the drone in meters
  - **LIDARDPoints:** LIDARPoints data is provided (and should also be the output format, if you generate LIDAR data) as a CSV file. The first line has the scan ID and number of data lines (number of recorded points for that sweep). Each following line has the angle of the data point (in degrees) and the distance (in millimeters) until the next scan ID header line. 34 sweeps are included.
  - **Mapping:** If you generate a map of the rooms (Task 5) the results should be printed to a csv file. Each line of the file should represent one wall in the building. Each wall should be represented by its start and end point in millimeters (xstart, ystart, xend, yend)

Guidelines:

- Expect to spend between 2-6 hours to produce a working solution.
- It is up to you to decide upon any unspecified parameters. Be creative!
- Feel free to use whichever language you are most comfortable in (Python, Java or C++ are preferable, Please only use freely available languages and compilers that are available on Linux or Windows 10.)
- Keep your code clean, organized and well documented !
- Essential: Include a README in your repo which explains your design decisions and instructions on how to compile and run your code.
- Good code includes test so we would expect to see some degree of testing included
- Please do not share the contents of this assignment publicly.
- Please commit regularly to your version control software, we'd like to see a progression of your software development not just an uploaded folder.

Evaluation: In particular some aspects we are looking for:

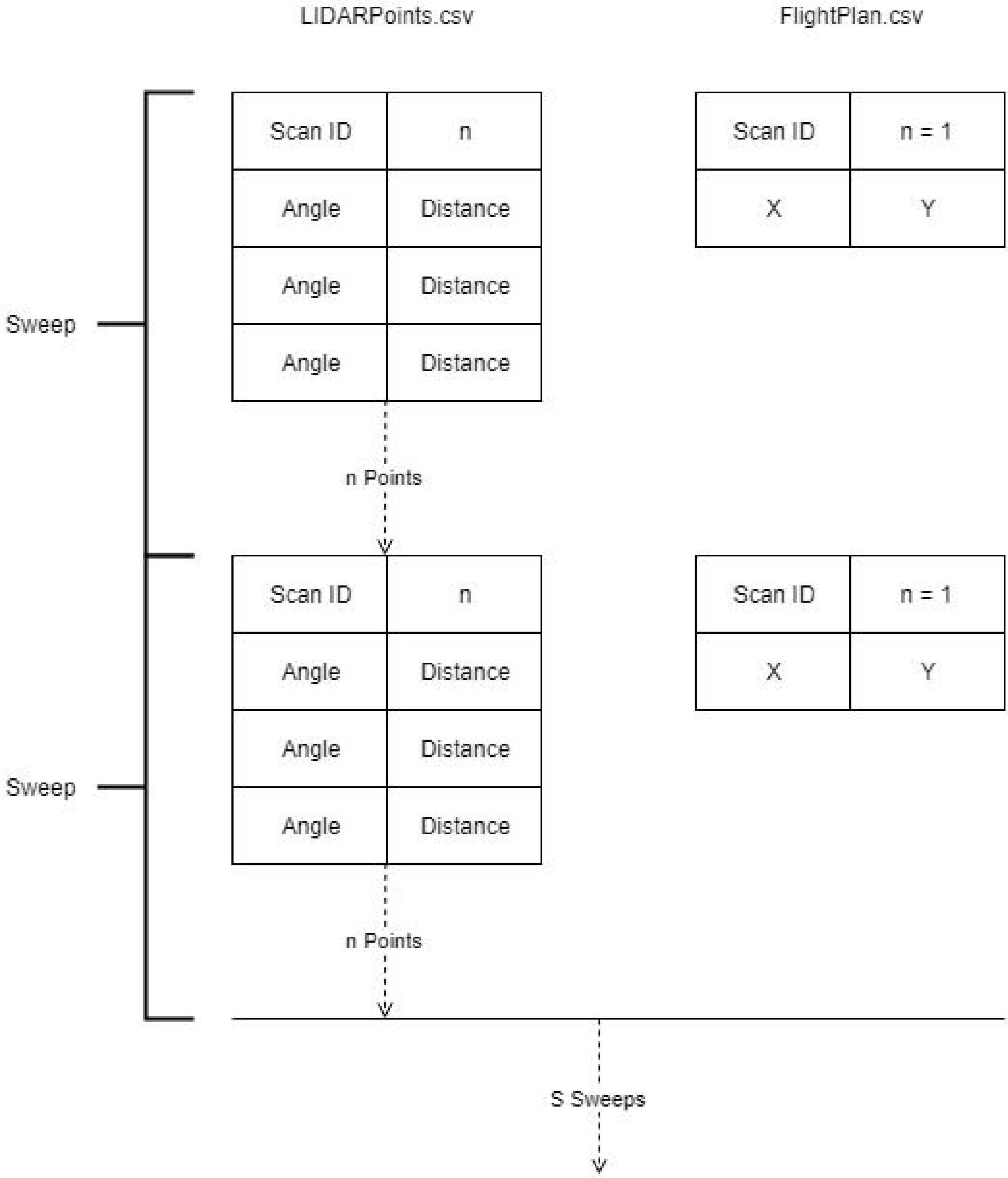
- Logical design of the software
- Well documented and tested code
- Code maintainability, extensibility, readability
- Efficiency is an added bonus, the priority being all the above

Submission:

- Please return your submission via github/bitbucket or any other web git host and providing the URL. Please commit regularly to the repo to show your path to a solution.
- Please include a readme file with detailed instructions on how to build, run and use the code for either a Windows 10 or Linux Ubuntu system. If we struggle to run your system it will reflect badly on the assignment. If we cannot get your code to run we will not ask you, we will simply assume it doesn't run.

If you have any questions, please do not hesitate to contact us.

Format Example:



LIDARPoints.csv

Scan ID	n
Angle	Distance
Angle	Distance
Angle	Distance

n Points

Scan ID	n
Angle	Distance
Angle	Distance
Angle	Distance

n Points

FlightPlan.csv

Scan ID	n = 1
X	Y

Scan ID	n = 1
X	Y