University of Ontario Institute of Technology

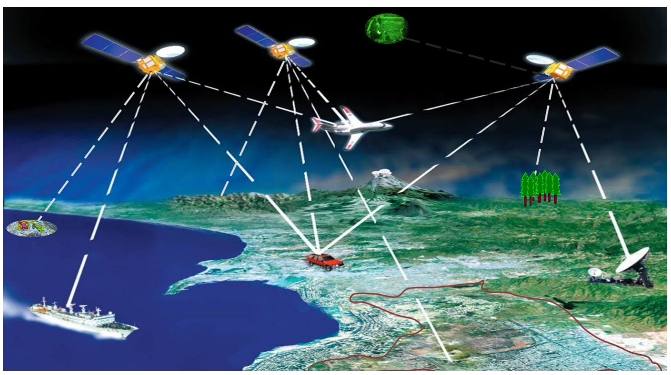
Programming Internal Engineering Competition 2017 Problem

GPS Dilemma

A GPS is connected to many satellites above earth providing a useful and accurate navigation system. Each satellite transmits a radio signal encoded with the time the signal was sent. Accurate clocks are within GPS receivers to be able to compare its local time with the encoded time from the radio signal of the satellites. Radio signals propagate at a known rate, because of this the GPS receiver can calculate the distance from it's current location to the location of the satellite when the signal was broadcast. The GPS receiver is able to calculate it's position accurately by measuring its distance from many satellites in known orbits.

The task that has been given to you is to write a basic "Autopilot" program for GPS navigation. It will be taken as a 2d problem, so do not take into account the curvature of the earth or the altitude of the satellites. This problem will be using speeds more suited for planes and soundwaves than for satellites and radio waves.

A set of signals will be sent from moving sources, the program must be able to compute the receiving position on the Cartesian Coordinate plane. Given a destination point on the Cartesian plane, the program must be then able to compute the heading angle required to go from the receiving position to the destination. The compass headings are all stated in degrees. Compass heading 0, or North corresponds to the positive y direction, whilst the compass heading 90, or East corresponds to the positive x direction, Just like it is shown in Figure 2.



Input:

The input consists of independent test sets each in it's own text file. You will be provided with 4 files for the purposes of developing and debugging the application. You will be evaluated using both these test cases, and ones that will be provided during the evaluation itself.

The first line of the input for each data set has an Integer N(1<=N<=10), this integer represents the number of signal sources in the set. This is followed by 3 floating point numbers: *t, x,* and *y.*

t is the local time exactly when all the signals have been received. The units for *t* is in seconds(s) after the reference time (0s). *x* and *y* are the coordinates of the destination point on the plane.

The next N lines have 4 floating point numbers that carry information about the Nth signal source. First 2 numbers are the known position of the signal source on the Cartesian plane at the reference time (t=0). Third number is the heading of the moving signal source (or satellite) in the form of a compass heading D (0<=D<360). Fourth number is the time that encoded within the signal (which is the time the signal was transmitted in seconds after the reference time). Magnitudes of each number in the input file are less than 10000 and floating-point numbers have max 5 places after the decimal point

The unit distance on the coordinate plane is 1 meter. Assume that each signal source is moving over the Cartesian plane at a speed of 100m/s and that the broadcast signal propagates with a speed of 350m/s. Due to inaccuracies in synchronizing clocks, assume that distance calculations are accurate only to 0.1 meter. So if 2 points are computed to be within 0.1 meter of each other, they should be treated as the same point. There is also a possibility that a signal may have been corrupted in transmission, so the data received from multiple signals may be inconsistent.

Output:

For each trial, display the compass heading from the receiving location to the destination, in degrees. If the signals do not contain enough information to compute the receiving location (where many locations are consistent with the signals) print “Inconclusive”. If the signals are inconsistent (no possible locations consistent with the signal), print “Inconsistent”.

User Interface:

Alongside the output, you can add some form of UI to it. Examples include making a visual of the compass pointing to the correct angle showcasing your solution, or displaying the signal sources moving, and the broadcasting signals converging on the location. These are just our suggestion, you don’t need to follow them, you can do what you think is best.

Figure 1: Compass

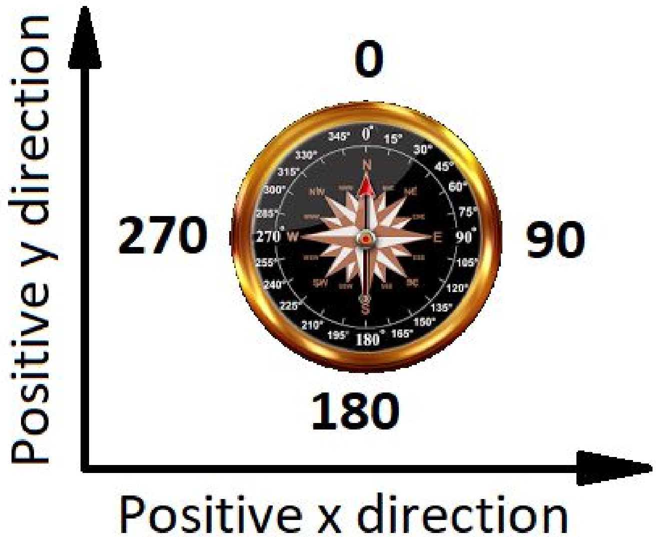


Figure 2: First Sample Input

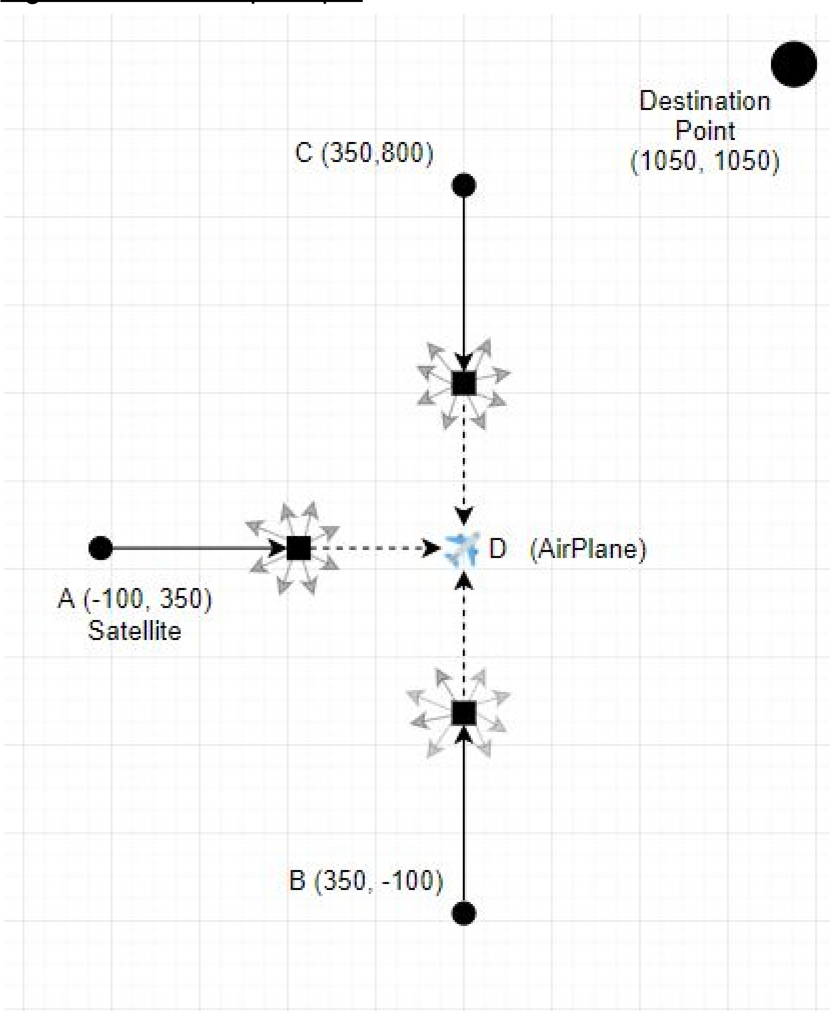


Diagram not to scale

Tasks:

- Solve for the solutions in the test cases.
- Incorporate user input alongside the test cases. Allow users to set N, t, x, y, and the 4 numbers within the ranges given, to allow them to calculate the angles.
- Write Pseudocode for your algorithm with comments

Small Bonus:

Do a Time Complexity analysis of the algorithm of your code.

Github Repository:

https://github.com/studiousKych/IEC_programming_2017_resources

Evaluation Criteria

Level	1	2	3	4
Algorithm	Makes some attempt to solve the question with no solution or logic	Makes an attempt using a brute force approach or only works with the given test cases	Successfully solves the problem, with different test cases, but is inefficient	Successfully solves the problem with an efficient algorithm with low complexity
Code Quality	Code is not commented, non-modular, unclear naming of variables.	Logical variable naming, commenting for main ideas.	Logical variable naming, comments are easy to follow, some use of functions and objects.	Logical variable naming, comments are easy to follow, completely modular.
Documentation	Plain text file, large sections missing.	Plain text file, covers main ideas.	A simple report covering main ideas in a word processor or a Slideshow/ power point	Detailed report with diagrams created with a documentation generator (Sphinx, Doxygen, etc.).
User Interface (UI)	Prints unformatted data to the command line.	Incorporates User input in a readable format	Incorporates User input in a user friendly format, along with some type of visual	Incorporates User input in a user friendly format, along with well-made visuals
Presentation	Unable to present or no presentation prepared, or unable to convey/explain their solution	Incomplete presentation prepared. Slightly able to explain their solution, unable to answer most questions	Complete simple presentation that explains their solution. Able to answer most questions	Presentation easy to follow with detail that explains their solution thoroughly. Presenters speak well and answer all questions correctly and well