Problem 12-E4 – The cosmic ship

It's 2123 and you're hired to write the soft for the first interstellar ship. Being an extremely complex system, you started by implementing the navigation system. In particular, there is a need for a program to direct the cosmic ship and to remember the distance traveled between the planets. The route will be calculated as follows: search for the closest unvisited planet and cover the distance to the planet, until it reaches all the detected planets. If there are several planets at the same distance from the current position, choose the first one from the input list.

Requirement

Having as input the space coordinates of the planes and the starting point, write a program that calculates the total distance traveled by the ship for visiting all the planets.

Input data

The program will receive, from the keyboard (*stdin* stream), on the first line an integer value *n* representing the number of planets. On the following *n* lines there are 3 fractional numbers, representing the *x*, *y* and *z* coordinates of the planets. On the last line there is the starting point of the ship in the same format as the planet coordinates. The coordinates are given in parcels from the center of the galaxy.

Output data

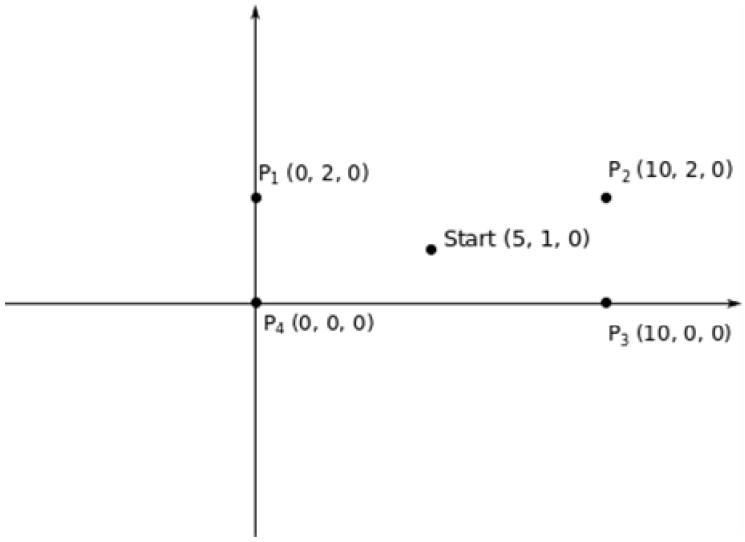
The program will display, on the output screen (the standard output stream), a single fractional value, with two decimals (obtained by rounding), representing the total distance in parcels traveled by the ship, until the last planet.

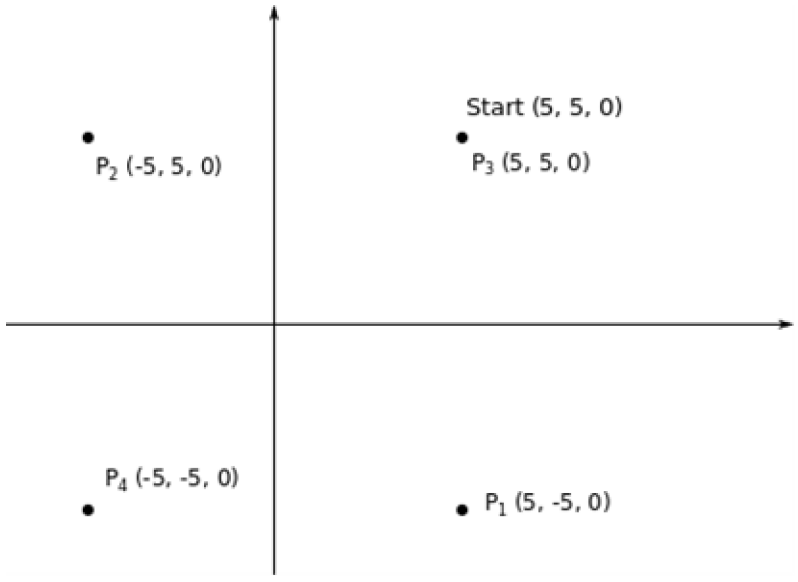
WARNING to the compliance of the problem requirement: displaying the results should be EXACTLY on the indicated way! In other words, the standard output stream will not show anything in addition to the requirement of the problem; as a result of automatic evaluation, any additional character displayed, or viewed other than that indicated, will lead to a false result and therefore obtain a Rejection of the program.

Restrictions and specifications

1. $0 < n <= 100$
2. The use of floating-point data types in double precision is recommended.
3. The calculations will be made at maximum accuracy, only the final result will be rounded for display purposes.
4. The *D* formula of the distance between two points $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ in a 3-dimensional space is: $D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$.
5. **Warning**: Depending on the chosen programming language, the file containing the code must have one of the following extensions .c, .cpp, .java, or .m. The web editor **will not automatically add** an extension and its absence leads to the impossibility of compiling the program!
6. **Warning**: The source file must be named, by the candidate, in the following form: <name>.<ext>, where name is the surname of the candidate and the extension is chosen according to the previous point. Pay attention to the limitations of the Java language, related to the class name and file name!

Examples

Input	Output	Explanation:
4 0 2 0 10 2 0 10 0 0 0 0 0 5 1 0	19.10	 From the starting point, all the planets are at the same distance, so choose the first one from the list, that is, P1. From Start to P1, the distance is 5.099019514. From P1, the closest planet is P4. The distance traveled becomes 7.099019514. From P4, the nearest unvisited planet is P3, the distance traveled becomes 17.099019514. Then the last move is from P3 to P2, the total distance being 19.099019514. Displayed with 2 decimals, the result is 19.10.

4 5 -5 0 -5 5 0 5 5 0 -5 -5 0 5 5 0	30.00	 The starting point coincides with P3 so the first stop is right there and the distance traveled is 0. Then, the nearest planet in the order of input is P1, the distance traveled being 10. From P1 the ship will travel to P4 and then to P2. The total distance is 30.

Working time: 120 minutes