Travelling Salesperson 2D

Input

Your program should take as input a single TSP instance. It will be run several times, once for every test case. The time limit is per test case.

The first line of standard input contains an integer $1 \leq N \leq 1000$, the number of points. The following N lines each contain a pair of real numbers x, y giving the coordinates of the N points. All numbers in the input have absolute value bounded by 10^6.

The distance between two points is computed as the Euclidean distance between the two points, *rounded to the nearest integer*.

Output

Standard output should consist of N lines, the i'th of which should contain the (0-based) index of the i'th point visited.

Score

Let Opt be the length of an optimal tour, Val be the length of the tour found by your solution, and Naive be the length of the tour found by the algorithm below. Define $x = (\mathrm{Val} - \mathrm{Opt})/(\mathrm{Naive} - \mathrm{Opt})$. (In the special case that $\mathrm{Naive} = \mathrm{Opt}$, define $x = 0$ if $\mathrm{Val} = \mathrm{Opt}$, and $x = \infty$ if $\mathrm{Val} > \mathrm{Opt}$.)

The score on a test case is 0.02^x. Thus, if your tour is optimal, you will get 1 point on this test case. If your score is halfway between Naive and Opt, you get roughly 0.14 points. The total score of your submission is the sum of your score over all test cases. There will be a total of 50 test cases. Thus, an implementation of the algorithm below should give a score of at least 1 (it will get 0.02 points on most test cases, and 1.0 points on some easy cases where it manages to find an optimal solution).

The algorithm giving the value Naive is as follows:

```
GreedyTour
   tour[0] = 0
   used[0] = true
   for i = 1 to n-1
      best = -1
      for j = 0 to n-1
         if not used[j] and (best = -1 or dist(tour[i-1],j) < dist(tour[i-1],best))
            best = j
      tour[i] = best
      used[best] = true
   return tour
```

The sample output gives the tour found by `GreedyTour` on the sample input. The length of the tour is 323.

Sample Input 1

```
10
95.0129 61.5432
23.1139 79.1937
60.6843 92.1813
48.5982 73.8207
89.1299 17.6266
76.2097 40.5706
45.6468 93.5470
1.8504 91.6904
82.1407 41.0270
44.4703 89.3650
```

Sample Output 1

```
0
8
5
4
3
9
6
2
1
7
```

Submit Stats

Problem ID: tsp

CPU Time limit: 2 seconds

Memory limit: 1024 MB

Download:
Sample data files

Author: Per Austrin

Source: KTH CSC Avancerade Algoritmer 2006