Traveling Salesman Problem (TSP) is an important problem in computer science. The TSP aims at finding a route passed all cities exactly once with minimum traveling distance. There are two representations for TSP problems: using the coordinates of all cities or using the distance matrix for the distance between all pairs of the cities.

The solution of a TSP instance can be represented as a permutation, which denotes the order of traveling these cities. A possible improvement with respect to the distance of a route can be found through systematically searching for all possible swaps for two cities' routing order. Such swap can be made on the permutation by simply swap the two values with different indices. Suppose the two indices are i and j, then we can go through all possible pairs of i and j by looping i and then looping j in increasing order.

Your goal is to write a program to store the TSP problem represented by coordinates of all cities, to translate (store) the representation to distance (cost) matrix, to store the a solution of a TSP instance, output the route and route length of such solution on the TSP instance, and output the improved route and route length based on the above method.

You should provide a class for the TSP solver, containing a TSP class object and a permutation array for the state. Overload the stream insertion and extraction operators for the classes TSP and TSP solver so that the input and output can be performed by the following statements:

```
cin >> tspSolverObject;
cout << tspSolverObject;
cin >> tspObject;
cout << tspObject;
```

Also provide a default constructor, copy constructor, assignment operator, and destructor for the TSP class, and provide a constructor with one argument of TSP object for the TSP solver class. Remember to use dynamic memory allocation in the class according to the input of number of cities.

Requirement: Provide classes to store TSP and construct a solver to make improvement on a given route. Prepare appropriate constructor and overloaded operators for your classes, and encapsulate the methods. Separate your program in files of three kinds: the class header file (.h), the class source code file (.cpp), and the file containing main function (.cpp).

Prohibited: Use C-style input/output.

Input

Each case contains an integer n (the number of cities), 2n integers (the coordinates of the n cities), and n integers (a possible solution for the TSP instance). The input ends with -1.

Output

For each case, output the original and improved route and route length in two lines. Each two consecutive cases are separated by the end line stream manipulator.

Sample Input

```
5
-1 2
3 1
4 3
5 -4
-5 0
3 1 2 4 0
-1
```

Sample Output

```
Route: 3 1 2 4 0 has distance: 28
Route: 2 1 3 4 0 has distance: 26
```