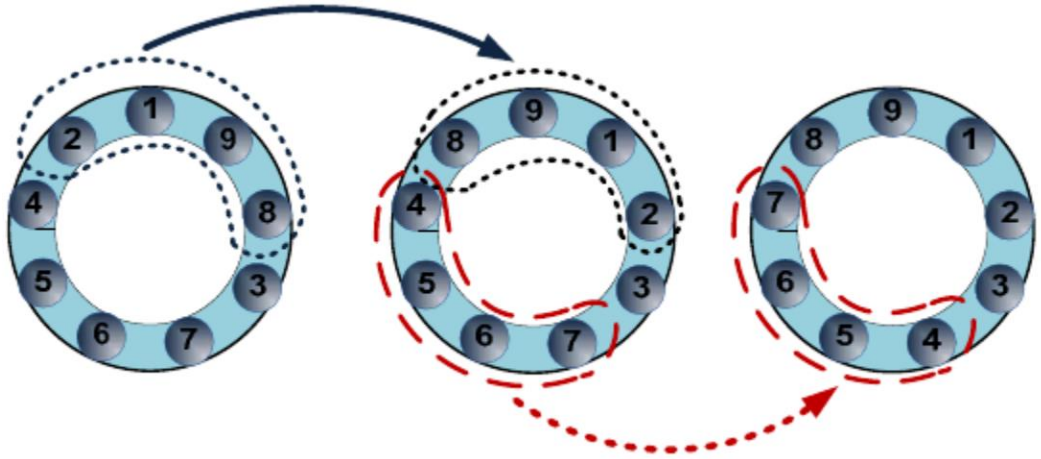
P9. Lazy Susan

(Time Limit: 3 seconds)

There are N marbles, which are labeled 1, 2, . . . , N. The N marbles are put in a circular track in an arbitrary order. In the top part of the track there is a “lazy Susan”, which is a tray that can hold exactly 4 marbles. The tray can be rotated, reversing the orientation of the four marbles. The tray can also be moved around the track in both directions.

For example, 9 marbles 1, 9, 8, 3, 7, 6, 5, 4, 2 are put in the circular track in clockwise order as shown in the following figure. This figure also shows how the tray is moved and rotated.



Trung wants you to arrange the marbles by moving and rotating the tray so that when listing the marbles from some position in the track in clockwise order, we get (1, 2, . . . , N). Your task is to write a program to tell Trung that either this can be done or not.

Input

The input file consists of several data sets. The first line of the input file contains the number of data sets which is a positive integer and is not bigger than 100. The following lines describe the data sets.

For each data set, the first line contains the integer N ($8 \le N \le 500$). The second line describes the initial state of the track. It contains N numbers which are the labels of the marbles when listing in clockwise order.

Output

For each test case, write in one line ‘possible’ if there exists a solution to arrange the marbles. If not so, write ‘impossible’.

Sample Input

```
2
9
1 9 8 3 7 6 5 4 2
11
```

1 3 2 4 5 6 7 8 9 10 11

Sample Input

possible

impossible

P10. Full Tank?

(Time Limit: 2 seconds)

After going through the receipts from your car trip through Europe this summer, you realised that the gas prices varied between the cities you visited. Maybe you could have saved some money if you were a bit more clever about where you filled your fuel?

To help other tourists (and save money yourself next time), you want to write a program for finding the cheapest way to travel between cities, filling your tank on the way. We assume that all cars use one unit of fuel per unit of distance, and start with an empty gas tank.

Input

The first line of input gives $1 \leq n \leq 1000$ and $0 \leq m \leq 10000$, the number of cities and roads. Then follows a line with n integers $1 \leq p_i \leq 100$, where p_i is the fuel price in the ith city. Then follow m lines with three integers $0 \leq u, v <$ n and $1 \leq d \leq 100$, telling that there is a road between u and v with length d. Then comes a line with the number $1 \leq q \leq 100$, giving the number of queries, and q lines with three integers $1 \leq c \leq 100$, s and e, where c is the fuel capacity of the vehicle, s is the starting city, and e is the goal.

Output

For each query, output the price of the cheapest trip from s to e using a car with the given capacity, or 'impossible' if there is no way of getting from s to e with the given car.

Sample Input

```
5 5
10 10 20 12 13
0 1 9
0 2 8
1 2 1
1 3 11
2 3 7
2
10 0 3
20 1 4
```

Sample Output

```
170
impossible
```