D
Fake scoreboard

As you know, after the award ceremony of SWERC it is customary to publish a complete scoreboard with detailed information on the submissions and verdicts received. However, due to the buggy contest management system, most of the relevant data are not being recorded today. Clearly such state of affairs fails to meet the high standards we are committed to, so the judges have resolved to make up the rest of the data based on whatever shred of information left, and hope contestants are unable to tell the difference. To make our lives even simpler, we kindly ask you to provide a solution for us, or else today's scoreboard will remain forever veiled in mystery (even the fake one).

What we will know by the end of the contest is the number T of teams, the number P of problems, and the number of accepted submissions by each team. From the number and colour of balloons floating around on the premises we will also be able to infer how many teams solved each of the problems. Your task is to figure out which teams solved which problems.

Our counting skills are not up to par, so your program should be able to detect when the data we collected must be wrong (see sample input 1). Otherwise you should output a possible solution, represented as a sequence of T strings of P characters each, in the following way. Both problems and teams are assigned with distinct integers, from 1 to P and 1 to T, respectively. For team number i ($1 \leq i \leq T$), write the string on alphabet `N,Y` such that its j-th ($1 \leq j \leq P$) character is `Y` if the team managed to get problem j accepted, and `N` otherwise. For example, the following three strings form a solution to the second sample case, where the score of each of three teams is $2, 1, 2$, and the count of accepted submissions for each of three problems is $1, 2, 2$:

```
NYY
NNY
YYN
```

There is at least one other solution, namely

```
NYY
NYN
YNY
```

When several solutions are possible we ask you to supply the one giving rise to the lexicographically smallest string, when each of the T rows are concatenated in order. In the example above we prefer the first solution, since `NYYNNYYYN` comes before `NYYNYNYNY` in lexicographical order. (String S comes before S' in lexicographical order if the first different character between the two is `N` in S but `Y` in S').

Input

Each input case is described by three lines:

The first contains two space-separated integers T (the number of teams) and P (the number of problems), with $1 \leq T, P \leq 80$. The second contains T space-separated integers between 0 and 90 (inclusive), the i-th of which indicates the number of problems solved by team i. The third (and last) line has P integers between 0 and 90, the j-th of which describes the number of teams successfully solving problem j.

Different input cases are separated by a blank line. The last line of the input file will be `0 0`.

Output

If the input data has a solution, print T lines of P characters each, depicting the lexicographically smallest solution as explained above. Otherwise output a single line with the word `Impossible`. In any case a blank line should separate outputs for different test cases.

Sample Input

```
2 2
1 2
1 1

3 3
2 1 2
1 2 2

3 5
3 3 1
3 1 1 0 2

0 0
```

Sample Output

```
Impossible

NYY
NNY
YYN

YNYNY
YYNNY
YNNNN
```