Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any Search Engine like Google, Yahoo, Bing ...

All the best for your test

Globussoft

QUESTION - 1

You are given two strings, A and B. Answer, what is the smallest number of operations you need to
transform A to B?

Operations are:

1) Delete one letter from one of strings
2) Insert one letter into one of strings
3) Replace one of letters from one of strings with another letter

Input

T - number of test cases

For each test case:

String A
String B

Both strings will contain only uppercase characters and they won't
be longer than 2000 characters.

There will be 10 test cases in data set.

Output

For each test case, one line, minimum number of operations.

Example

Input:

1
FOOD
MONEY

Output:

4

QUESTION – 2

Jane and Tarzan have got cellphones and now they want to be available all the time. They live in a long line of trees and they are available iff the absolute difference between the heights of the two trees which they are on is not greater than D.

Jane and Tarzan are moving by the following rules: each second both Jane and Tarzan (simultaneously) jump, from the tree which they are on, on one of the adjacent trees (the left or the right one). It is forbidden to stay on the same tree. Moreover, in the very beginning and after each second they have to be available.

The trees in the line are numbered from 1 to N, respectively. Jane is interested in the pairs of trees (J, T) – let's call them good pairs – for which it holds: if Jane begins on the J-th tree, and Tarzan on the T-th tree, they can swap their positions after some time (moving by the rules) – so that Tarzan ends on the J-th tree, and Jane on the T-th tree.

Have a look on the first test example. D = 0 means that the heights of the trees which Jane and Tarzan are on have to differ by at most 0 (therefore, have to be equal) all the time. Pair (1, 5) is a good pair since we can take 1-2-3-4-5-6-5 as Jane's route, and 5-6-5-4-3-2-1 as Tarzan's route – this results in Jane and Tarzan swapping their initial positions (and being available all the time).

Output all the good pairs (J, T) in which J < T.

Input

In the first line of input there are integers N ($1 \leq N \leq 100\,000$) and D ($0 \leq D \leq 10^9$).

In the next N lines there are N natural numbers less than 10^9 – heights of the trees in the line (from the 1st tree to the N-th tree).

Output

Output all the requested pairs in sorted order. We define a pair (A, B) to be smaller than the pair (C, D) iff (A < C) or (A = C and B < D).

In all of the test data, the number of these pairs will not exceed 100 000.

Example

```
Input:
6 0
2
1
2
3
2
1
```

Output:
1 3
1 5
2 6
3 5
Input:
5 10
10
20
10
5
10

Output:
1 2
1 3
1 4
2 3
2 5
3 4
3 5
4 5

QUESTION – 3

The human tribe has just discovered some other tribe and wants to communicate with them. To make sure it is not intercepted by the terminators, they ask their chief computer engineer Rohit to design a system for the purpose. In the design that Rohit proposes, data is transmitted n times. If it is received more than half-the times, it is said to be successfully transmitted. If not, the data is said to be lost. Rohit obviously got a lot of fame and respect for his work. Nitish doesn't like it and wants to challenge Rohit's supremacy. He wants to check out the system and has hired you for the process.

Input

The first line of the input contains test cases t(1<=t<=100). It is followed by 2*t lines, 2 for each test case. The first line of input for each test case contains a number n ($0<=n<=10^6$), followed by n elements in the next line. Each number is from -10^3 to +10^3

Output

You are required to output 'YES' followed by the number transmitted, if it was transmitted successfully, and 'NO' otherwise.

Example

Input:

```
3

4

2 1 2 2

6

1 1 1 2 2 2

5

1 2 4 5 1
```

Output:

```
YES 2

NO

NO
```

QUESTION – 4

n fish, numbered from 1 to *n*, live in a lake. Every day right one pair of fish meet, and the probability of each other pair meeting is the same. If two fish with indexes i and j meet, the first will eat up the second with the probability a_{ij}, and the second will eat up the first with the probability $a_{ji} = 1 - a_{ij}$. The described process goes on until there are at least two fish in the lake. For each fish find out the probability that it will survive to be the last in the lake.

Input

The first line contains integer n ($1 \le n \le 18$) — the amount of fish in the lake. Then there follow n lines with n real numbers each — matrix a. a_{ij} ($0 \le a_{ij} \le 1$) — the probability that fish with index i eats up fish with index j. It's guaranteed that the main diagonal contains zeros only, and for other elements the following is true: $a_{ij} = 1 - a_{ji}$. All real numbers are given with not more than 6 characters after the decimal point.

Output

Output n space-separated real numbers accurate to not less than 6 decimal places. Number with index i should be equal to the probability that fish with index i will survive to be the last in the lake.

Example

```
Input:
5
0 1 1 1 1
0 0 0.5 0.5 0.5
0 0.5 0 0.5 0.5
0 0.5 0.5 0 0.5
0 0.5 0.5 0.5 0
```

```
Output:

1.000000 0.000000 0.000000 0.000000 0.000000
```

QUESTION – 5

The set of cyclic rotations of a string are the strings obtained by embedding the string clockwise on a ring, with the first character following on the last, starting at any character position and moving clockwise on the ring until the character preceeding the starting character is reached. A string is a necklace if it is the lexicographically smallest among all its cyclic rotations. For instance, for the string 01011 the cyclic rotations are (10110,01101,11010,10101,01011), and furthermore 01011 is the smallest string and hence, a necklace.

Any string S can be written in a unique way as a concatenation S = T1 T2 . . . Tk of necklaces Ti such that Ti+1 < Ti for all i = 1, . . . , k − 1, and Ti Ti+1 is not a necklace for any i = 1, . . . , k − 1. This representation is called the necklace decomposition of the string S, and your task is to find it.
The relation < on two strings is the lexicographical order and has the usual interpretation: A < B if A is a proper prefix of B or if A is equal to B in the first j − 1 positions but smaller in the jth position for some j. For instance, 001 < 0010 and 1101011 < 1101100.

Input

On the first line of the input is a single positive integer n, telling the number of test scenarios to follow. Each scenario consists of one line containing a non-empty string of zeros and ones of length at most 100.

Output

For each scenario, output one line containing the necklace decomposition of the string. The necklaces should be written as ’(’ necklace ’)’.

Example

```
Input:
5
```

```
0
0101
0001
0010
11101111011
```

Output:

```
(0)
(0101)
(0001)
(001)(0)
(111)(01111)(011)
```