A. Game Outcome

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sherlock Holmes and Dr. Watson played some game on a checkered board $n \times n$ in size. During the game they put numbers on the board's squares by some tricky rules we don't know. However, the game is now over and each square of the board contains exactly one number. To understand who has won, they need to count the number of *winning* squares. To determine if the particular square is winning you should do the following. Calculate the sum of all numbers on the squares that share this column (including the given square) and separately calculate the sum of all numbers on the squares that share this row (including the given square). A square is considered *winning* if the sum of the column numbers is **strictly** greater than the sum of the row numbers.

For instance, lets game was ended like is shown in the picture. Then the purple cell is winning, because the sum of its column numbers equals $8+3+6+7=24$, sum of its row numbers equals $9+5+3+2=19$, and $24>19$.

Input

The first line contains an integer n ($1 \le n \le 30$). Each of the following n lines contain n space-separated integers. The j-th number on the i-th line represents the number on the square that belongs to the j-th column and the i-th row on the board. All number on the board are integers from 1 to 100.

Output

Print the single number — the number of the winning squares.

Examples

input
```
1
1
```
output
```
0
```

input
```
2
1 2
3 4
```
output
```
2
```

input
```
4
5 7 8 4
9 5 3 2
1 6 6 4
9 5 7 3
```
output
```
6
```

Note

In the first example two upper squares are winning.

In the third example three left squares in the both middle rows are winning:

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

B. Trace

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day, as Sherlock Holmes was tracking down one very important criminal, he found a wonderful painting on the wall. This wall could be represented as a plane. The painting had several concentric circles that divided the wall into several parts. Some parts were painted red and all the other were painted blue. Besides, any two neighboring parts were painted different colors, that is, the red and the blue color were alternating, i. e. followed one after the other. The outer area of the wall (the area that lied outside all circles) was painted blue. Help Sherlock Holmes determine the total area of red parts of the wall.

Let us remind you that two circles are called concentric if their centers coincide. Several circles are called concentric if any two of them are concentric.

Input

The first line contains the single integer n ($1 \le n \le 100$). The second line contains n space-separated integers r_i ($1 \le r_i \le 1000$) — the circles' radii. It is guaranteed that all circles are different.

Output

Print the single real number — total area of the part of the wall that is painted red. The answer is accepted if absolute or relative error doesn't exceed 10^{-4}.

Examples

input

```
1
1
```

output

```
3.1415926536
```

input

```
3
1 4 2
```

output

```
40.8407044967
```

Note

In the first sample the picture is just one circle of radius 1. Inner part of the circle is painted red. The area of the red part equals $\pi \times 1^2 = \pi$.

In the second sample there are three circles of radii 1, 4 and 2. Outside part of the second circle is painted blue. Part between the second and the third circles is painted red. Part between the first and the third is painted blue. And, finally, the inner part of the first circle is painted red. Overall there are two red parts: the ring between the second and the third circles and the inner part of the first circle. Total area of the red parts is equal $(\pi \times 4^2 - \pi \times 2^2) + \pi \times 1^2 = \pi \times 12 + \pi = 13\pi$

C. Message

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dr. Moriarty is about to send a message to Sherlock Holmes. He has a string s.

String p is called a *substring* of string s if you can read it starting from some position in the string s. For example, string "aba" has six substrings: "a", "b", "a", "ab", "ba", "aba".

Dr. Moriarty plans to take string s and cut out some substring from it, let's call it t. Then he needs to *change* the substring t zero or more times. As a result, he should obtain a fixed string u (which is the string that should be sent to Sherlock Holmes). One change is defined as making one of the following actions:

- Insert one letter to any end of the string.
- Delete one letter from any end of the string.
- Change one letter into any other one.

Moriarty is very smart and after he chooses some substring t, he always makes the minimal number of changes to obtain u.

Help Moriarty choose the best substring t from all substrings of the string s. The substring t should minimize the number of changes Moriarty should make to obtain the string u from it.

Input

The first line contains a non-empty string s, consisting of lowercase Latin letters. The second line contains a non-empty string u, consisting of lowercase Latin letters. The lengths of both strings are in the range from 1 to 2000, inclusive.

Output

Print the only integer — the minimum number of changes that Dr. Moriarty has to make with the string that you choose.

Examples

input

```
aaaaa
aaa
```

output

```
0
```

input

```
abcabc
bcd
```

output

```
1
```

input

```
abcdef
klmnopq
```

output

```
7
```

Note

In the first sample Moriarty can take any substring of length 3, and it will be equal to the required message u, so Moriarty won't have to make any changes.

In the second sample you should take a substring consisting of characters from second to fourth ("bca") or from fifth to sixth ("bc"). Then you will only have to make one change: to change or to add the last character.

In the third sample the initial string s doesn't contain any character that the message should contain, so, whatever string you choose, you will have to make at least 7 changes to obtain the required message.

D. Suspects

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

As Sherlock Holmes was investigating a crime, he identified n suspects. He knows for sure that exactly one of them committed the crime. To find out which one did it, the detective lines up the suspects and numbered them from 1 to n. After that, he asked each one: "Which one committed the crime?". Suspect number i answered either "The crime was committed by suspect number a_i", or "Suspect number a_i didn't commit the crime". Also, the suspect could say so about himself ($a_i = i$).

Sherlock Holmes understood for sure that exactly m answers were the truth and all other answers were a lie. Now help him understand this: which suspect lied and which one told the truth?

Input

The first line contains two integers n and m ($1 \le n \le 10^5, 0 \le m \le n$) — the total number of suspects and the number of suspects who told the truth. Next n lines contain the suspects' answers. The i-th line contains either "+a_i" (without the quotes), if the suspect number i says that the crime was committed by suspect number a_i, or "-a_i" (without the quotes), if the suspect number i says that the suspect number a_i didn't commit the crime (a_i is an integer, $1 \le a_i \le n$).

It is guaranteed that at least one suspect exists, such that if he committed the crime, then exactly m people told the truth.

Output

Print n lines. Line number i should contain "`Truth`" if suspect number i has told the truth for sure. Print "`Lie`" if the suspect number i lied for sure and print "`Not defined`" if he could lie and could tell the truth, too, depending on who committed the crime.

Examples

input
```
1 1
+1
```
output
```
Truth
```

input
```
3 2
-1
-2
-3
```
output
```
Not defined
Not defined
Not defined
```

input
```
4 1
+2
-3
+4
-1
```
output
```
Lie
Not defined
Lie
Not defined
```

Note

The first sample has the single person and he confesses to the crime, and Sherlock Holmes knows that one person is telling the truth. That means that this person is telling the truth.

In the second sample there are three suspects and each one denies his guilt. Sherlock Holmes knows that only two of them are telling the truth. Any one of them can be the criminal, so we don't know for any of them, whether this person is telling the truth or not.

In the third sample the second and the fourth suspect defend the first and the third one. But only one is telling the truth, thus, the first or the third one is the criminal. Both of them can be criminals, so the second and the fourth one can either be lying or telling the truth. The first and the third one are lying for sure as they are blaming the second and the fourth one.

E. Cipher

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sherlock Holmes found a mysterious correspondence of two VIPs and made up his mind to read it. But there is a problem! The correspondence turned out to be encrypted. The detective tried really hard to decipher the correspondence, but he couldn't understand anything.

At last, after some thought, he thought of something. Let's say there is a word s, consisting of $|s|$ lowercase Latin letters. Then for one *operation* you can choose a certain position p $(1 \le p < |s|)$ and perform one of the following actions:

- either replace letter s_p with the one that alphabetically **follows** it and replace letter s_{p+1} with the one that alphabetically **precedes** it;
- or replace letter s_p with the one that alphabetically **precedes** it and replace letter s_{p+1} with the one that alphabetically **follows** it.

Let us note that letter "z" doesn't have a defined following letter and letter "a" doesn't have a defined preceding letter. That's why the corresponding changes are not acceptable. If the operation requires performing at least one unacceptable change, then such operation cannot be performed.

Two words *coincide in their meaning* iff one of them can be transformed into the other one as a result of zero or more operations.

Sherlock Holmes needs to learn to quickly determine the following for each word: how many words can exist that coincide in their meaning with the given word, but differs from the given word in at least one character? Count this number for him modulo 1000000007 $(10^9 + 7)$.

Input

The input data contains several tests. The first line contains the only integer t $(1 \le t \le 10^4)$ — the number of tests.

Next t lines contain the words, one per line. Each word consists of lowercase Latin letters and has length from 1 to 100, inclusive. Lengths of words can differ.

Output

For each word you should print the number of different **other** words that coincide with it in their meaning — not from the words listed in the input data, but from all possible words. As the sought number can be very large, print its value modulo 1000000007 $(10^9 + 7)$.

Examples

input

```
1
ab
```

output

```
1
```

input

```
1
aaaaaaaaaaa
```

output

```
0
```

input

```
2
ya
klmbfxzb
```

output

```
24
320092793
```

Note

Some explanations about the *operation*:

- Note that for each letter, we can clearly define the letter that follows it. Letter "b" alphabetically follows letter "a", letter "c" follows letter "b", ..., "z" follows letter "y".
- Preceding letters are defined in the similar manner: letter "y" precedes letter "z", ..., "a" precedes letter "b".
- Note that the operation never changes a word's length.

In the first sample you can obtain the only other word "ba". In the second sample you cannot obtain any other word, so the correct answer is 0.

Consider the third sample. One operation can transform word "klmbfxzb" into word "klmcexzb": we should choose $p = 4$, and replace the fourth letter with the following one ("b" $\rightarrow$ "c"), and the fifth one — with the preceding one ("f" $\rightarrow$ "e"). Also, we can obtain many other words from this one. An operation can transform word "ya" only into one other word "xb".

Word "ya" coincides in its meaning with words "xb", "wc", "vd", ..., "ay" (overall there are 24 other words). The word "klmbfxzb has many more variants — there are 3320092814 other words that coincide with in the meaning. So the answer for the first word equals 24 and for the second one equals 320092793 — the number 3320092814 modulo $10^9 + 7$