A. Line Breaks

```
Input file:     standard input
Output file:    standard output
Time limit:     1 second
Memory limit:   256 megabytes
```

Kostya has a text s consisting of n words made up of Latin alphabet letters. He also has two strips on which he must write the text. The first strip can hold m characters, while the second can hold as many as needed.

Kostya must choose a number x and write the first x words from s on the first strip, while all the remaining words are written on the second strip. To save space, the words are written without gaps, but each word must be entirely on one strip.

Since space on the second strip is very valuable, Kostya asks you to choose the maximum possible number x such that all words $s_1, s_2, \dots, s_x$ fit on the first strip of length m.

Input

The first line contains an integer t $(1 \leq t \leq 1000)$ — the number of test cases.

The first line of each test case contains two integers n and m $(1 \leq n \leq 50; 1 \leq m \leq 500)$ — the number of words in the list and the maximum number of characters that can be on the first strip.

The next n lines contain one word s_i of lowercase Latin letters, where the length of s_i does not exceed 10.

Output

For each test case, output the maximum number of words x such that the first x words have a total length of no more than m.

Standard Input	Standard Output
5 3 1 a b c 2 9 alpha beta 4 12 hello world and codeforces 3 2 ab c d 3 2 abc ab a	1 2 2 1 0

,

B. Transfusion

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given an array a of length n. In one operation, you can pick an index i from 2 to $n-1$ inclusive, and do one of the following actions:

- Decrease a_{i-1} by 1, then increase a_{i+1} by 1.
- Decrease a_{i+1} by 1, then increase a_{i-1} by 1.

After each operation, all the values must be non-negative. Can you make all the elements equal after any number of operations?

Input

First line of input consists of one integer t $(1 \le t \le 10^4)$ — the number of test cases.

First line of each test case consists of one integer n ($3 \le n \le 2 \cdot 10^5$).

Second line of each test case consists of n integers a_i ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n of all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print "YES" without quotation marks if it is possible to make all the elements equal after any number of operations; otherwise, print "NO" without quotation marks.

You can print answers in any register: "yes", "YeS", "nO" — will also be considered correct.

Standard Input	Standard Output

8
3
3 2 1
3
1 1 3
4
1 2 5 4
4
1 6 6 1
5
6 2 1 4 2
4
1 4 2 1
5
3 1 2 1 3
3
2 4 2
	YES
	NO
	YES
	NO
	YES
	NO
	NO
	NO

C. Uninteresting Number

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given a number n with a length of no more than 10^5.

You can perform the following operation any number of times: choose one of its digits, square it, and replace the original digit with the result. The result must be a digit (that is, if you choose the digit x, then the value of x^2 must be less than 10).

Is it possible to obtain a number that is divisible by 9 through these operations?

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.

The only line of each test case contains the number n, without leading zeros. The length of the number does not exceed 10^5.

It is guaranteed that the sum of the lengths of the numbers across all test cases does not exceed 10^5.

Output

For each test case, output "YES" if it is possible to obtain a number divisible by 9 using the described operations, and "NO" otherwise.

You can output each letter in any case (lowercase or uppercase). For example, the strings "yEs", "yes", "Yes", and "YES" will be accepted as a positive answer.

Standard Input	Standard Output
9 123 322 333333333333 9997 5472778912773 1234567890 23 33 52254522632	NO YES YES NO NO YES NO YES YES

Note

In the first example, from the integer 123, it is possible to obtain only 123, 143, 129, and 149, none of which are divisible by 9.

In the second example, you need to replace the second digit with its square; then n will equal $342 = 38 \cdot 9$.

In the third example, the integer is already divisible by 9.

D. Digital string maximization

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a string s, consisting of digits from 0 to 9. In one operation, you can pick any digit in this string, except for 0 or the leftmost digit, decrease it by 1, and then swap it with the digit left to the picked.

For example, in one operation from the string 1023, you can get 1103 or 1022.

Find the lexicographically maximum string you can obtain after any number of operations.

Input

The first line of the input consists of an integer t $(1 \leq t \leq 10^4)$ — the number of test cases.

Each test case consists of a single line consisting of a digital string s $(1 \leq |s| \leq 2 \cdot 10^5)$, where $|s|$ denotes the length of s. The string does not contain leading zeroes.

It is guaranteed that the sum of $|s|$ of all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, print the answer on a separate line.

Standard Input	Standard Output
6 19 1709 11555 51476 9876543210 5891917899	81 6710 33311 55431 9876543210 7875567711

Note

In the first example, the following sequence of operations is suitable: $19 \rightarrow 81$.

In the second example, the following sequence of operations is suitable: $1709 \rightarrow 1780 \rightarrow 6180 \rightarrow 6710$.

In the fourth example, the following sequence of operations is suitable:
$51476 \rightarrow 53176 \rightarrow 53616 \rightarrow 53651 \rightarrow 55351 \rightarrow 55431$.

E. Three Strings

```
Input file:     standard input
Output file:    standard output
Time limit:     2.5 seconds
Memory limit:   256 megabytes
```

You are given three strings: a, b, and c, consisting of lowercase Latin letters. The string c was obtained in the following way:

1. At each step, either string a or string b was randomly chosen, and the first character of the chosen string was removed from it and appended to the end of string c, until one of the strings ran out. After that, the remaining characters of the non-empty string were added to the end of c.
2. Then, a certain number of characters in string c were randomly changed.

For example, from the strings $a = \mathrm{abra}$ and $b = \mathrm{cada}$, without character replacements, the strings $\mathrm{caabdraa}$, $\mathrm{abracada}$, $\mathrm{acadabra}$ could be obtained.

Find the minimum number of characters that could have been changed in string c.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^3$) — the number of test cases.

The first line of each test case contains one string of lowercase Latin letters a ($1 \le |a| \le 10^3$) — the first string, where $|a|$ denotes the length of string a.

The second line of each test case contains one string of lowercase Latin letters b ($1 \le |b| \le 10^3$) — the second string, where $|b|$ denotes the length of string b.

The third line of each test case contains one string of lowercase Latin letters c ($|c| = |a| + |b|$) — the third string.

It is guaranteed that the sum of $|a|$ across all test cases does not exceed $2 \cdot 10^3$. Also, the sum of $|b|$ across all test cases does not exceed $2 \cdot 10^3$.

Output

For each test case, output a single integer — the minimum number of characters that could have been changed in string c.

Standard Input	Standard Output
7	1
a	0
b	2
cb	0
ab	3
cd	2
acbd	3
ab	
ba	
aabb	
xxx	
yyy	

xyxyxy a bcd decf codes horse codeforces egg annie egaegaeg	

F. Maximum modulo equality

```
Input file:     standard input
Output file:    standard output
Time limit:     5 seconds
Memory limit:   256 megabytes
```

You are given an array a of length n and q queries l, r.

For each query, find the maximum possible m, such that all elements a_l, a_{l+1}, ..., a_r are equal modulo m. In other words, $a_l \bmod m = a_{l+1} \bmod m = \dots = a_r \bmod m$, where $a \bmod b$ — is the remainder of division a by b. In particular, when m can be infinite, print 0.

Input

The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases.

The first line of each test case contains two integers n, q $(1 \le n, q \le 2 \cdot 10^5)$ — the length of the array and the number of queries.

The second line of each test case contains n integers a_i $(1 \le a_i \le 10^9)$ — the elements of the array.

In the following q lines of each test case, two integers l, r are provided $(1 \le l \le r \le n)$ — the range of the query.

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$, and the sum of q does not exceed $2 \cdot 10^5$.

Output

For each query, output the maximum value m described in the statement.

Standard Input	Standard Output

3
5 5
5 14 2 6 3
4 5
1 4
2 4
3 5
1 1
1 1
7
1 1
3 2
1 7 8
2 3
1 2
	3 1 4 1 0
	0
	1 6

Note

In the first query of the first sample, $6 \bmod 3 = 3 \bmod 3 = 0$. It can be shown that for greater m, the required condition will not be fulfilled.

In the third query of the first sample, $14 \bmod 4 = 2 \bmod 4 = 6 \bmod 4 = 2$. It can be shown that for greater m, the required condition will not be fulfilled.

,

G. Tree Destruction

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Given a tree[*] with n vertices. You can choose two vertices a and b once and remove all vertices on the path from a to b, including the vertices themselves. If you choose $a = b$, only one vertex will be removed.

Your task is to find the maximum number of connected components[†] that can be formed after removing the path from the tree.

[*] A tree is a connected graph without cycles.

[†] A connected component is a set of vertices such that there is a path along the edges from any vertex to any other vertex in the set (and it is not possible to reach vertices not belonging to this set)

Input

The first line of the input contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the size of the tree.

The next $n - 1$ lines contain two integers u and v ($1 \le u, v \le n$, $u \ne v$) — the vertices connected by an edge. It is guaranteed that the edges form a tree.

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output one integer — the maximum number of connected components that can be achieved using the described operation.

Standard Input	Standard Output

6
2
1 2
5
1 2
2 3
3 4
3 5
4
1 2
2 3
3 4
5
2 1
3 1
4 1
5 4
6
2 1
	1
	3
	2
	3
	4
	3

3 1
4 1
5 3
6 3
6
2 1
3 2
4 2
5 3
6 4	