A. Contest Performance

time limit per test: 2 s.
memory limit per test: 256 MB

In the Mind Hack Contest, n programmers participated. Each programmer solved a certain number of problems within the given time — this number reflects their performance in the contest.

After the contest ended, each programmer started wondering:

"What is the difference between the number of problems I solved and the highest number of problems solved by any other programmer?"

Rules:

- Each programmer excludes themselves from the comparison
- The difference can be positive if they solved more than others, or negative if someone else solved more

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 1000$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($2 \le n \le 2 \cdot 10^5$) — the number of programmers.

The following line contains n space-separated positive integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le 10^9$) — the number of problems solved by each programmer.

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

Output

For each test case, output n space-separated integers. For each i ($1 \le i \le n$) output the difference between s_i and the maximum number of problems solved by any other programmer.

Example

input

```
5
4
4 7 3 5
2
1 2
5
1 2 3 4 5
3
4 9 4
4
4 4 4 4
```

output

```
-3 2 -4 -2
-1 1
-4 -3 -2 -1 1
-5 5 -5
0 0 0 0
```

Note

For the first test case:

Number of problems solved by the programmers: [4 , 7 , 3 , 5]

- Programmer 1: $4 \rightarrow \max \text{others} = 7 \rightarrow \text{diff} = -3$
- Programmer 2: $7 \rightarrow \max \text{others} = 5 \rightarrow \text{diff} = 2$
- Programmer 3: $3 \rightarrow \max \text{others} = 7 \rightarrow \text{diff} = -4$
- Programmer 4: $5 \rightarrow \max \text{others} = 7 \rightarrow \text{diff} = -2$

Final Result: [-3 , 2 , -4 , -2]

B. Primo stamina

time limit per test: 5 s.
memory limit per test: 256 MB

Primo begins his stamina training. There are n exercise stations available, and the i-th station ($1 \le i \le n$) consists of a_i identical workout sets.

Given an integer u ($1 \le u \le 10^9$), completing each set changes Primo's stamina as follows:

- Completing the 1-st set increases stamina by u.
- Completing the 2-nd set increases stamina by $u - 1$.
- Completing the 3-rd set increases stamina by $u - 2$.
- ...
- Completing the k-th set ($k \ge 1$) increases stamina by $u + 1 - k$. (This value can be negative, which means doing extra sets decreases stamina.)

You are also given an integer l. You must choose an integer r such that $l \le r \le n$, and Primo will perform **all** sets in stations $l, l+1, \ldots, r$ (that is, a total of $\sum_{i=l}^{r} a_i = a_l + a_{l+1} + \ldots + a_r$ sets).

Answer the following: what is the optimal r you can choose so that the stamina increase is maximized?

If multiple r maximize the stamina increase, output the **smallest** r.

To increase the difficulty, you must answer this question for q different values of l and u.

Input

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

Each test case is described as follows:

- The first line contains an integer n ($1 \le n \le 10^5$).
- The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^4$).
- The third line contains an integer q ($1 \le q \le 10^5$).
- The next q lines each contain two integers l and u ($1 \le l \le n, 1 \le u \le 10^9$).

The sum of n across all test cases does not exceed $2 \cdot 10^5$. The sum of q across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output q integers: the i-th integer must be the optimal r for the i-th query. If there are multiple solutions, output the **smallest** one.

Examples

input

```
5
6
3 1 4 1 5 9
3
1 8
2 7
5 9
1
10
1
1 1
9
5 10 9 6 8 3 10 7 3
5
8 56
1 12
9 3
1 27
5 45
5
7 9 2 5 2
10
1 37
2 9
3 33
4 32
4 15
2 2
4 2
2 19
3 7
2 7
10
9 1 6 7 6 3 10 7 3 10
5
10 43
3 23
9 3
6 8
5 14
```

output

```
3 4 5
1
9 2 9 4 9
5 2 5 5 5 2 4 5 4 2
10 6 9 7 7
```

Note

For the 1-st query in the first test case:

- Choosing $r = 3$: Primo completes $a_1 + a_2 + a_3 = 3 + 1 + 4 = 8$ sets, stamina $= u + (u - 1) + \cdots + (u - 7) = 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 36$.
- Choosing $r = 4$: Primo completes $a_1 + a_2 + a_3 + a_4 = 9$ sets, stamina $= u + (u - 1) + \cdots + (u - 8) = 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0 = 36$.

Both choices give the same maximum, but we pick the smaller $r = 3$.
For the 2-nd query: choosing $r = 4$ gives the maximum stamina $= 27$.

For the 3-rd query: both $r = 5$ and $r = 6$ give the same stamina $= 35$, so we pick the smaller $r = 5$.

Thus, the output for the first test case is $[3, 4, 5]$.

C. Different One

time limit per test: 1 s.

memory limit per test: 256 MB

You are given three digits a, b, c. Two of them are equal, but the third one is different from the other two.

Find the value that occurs exactly once.

Input

The first line contains a single integer t ($1 \le t \le 270$) — the number of test cases.

The only line of each test case contains three digits a, b, c ($0 \le a, b, c \le 9$). Two of the digits are equal, but the third one is different from the other two.

Output

For each test case, output the value that occurs exactly once.

Example

input Copy

```
10
1 2 2
4 3 4
5 5 6
7 8 8
9 0 9
3 6 3
2 8 2
5 7 7
7 7 5
```

5 7 5

output Copy

```
1
3
6
7
0
6
8
5
5
7
```

D. Well Structured Sequence

time limit per test: 2 s.

memory limit per test: 256 MB

You are given a sequence of integers a of length n.

We call a sequence *well-structured* if it consists of consecutive blocks, where each block starts with its length followed by that many elements. For example, the sequences $[3, 3, 4, 5, 2, 6, 1]$ and $[1, 8, 4, 5, 2, 6, 1]$ are *well-structured* (blocks are colored differently), whereas $[1]$, $[1, 4, 3]$, $[3, 2, 1]$ are not.

In one move, you are allowed to remove any element from the sequence. Determine the minimum number of removals needed to transform the given sequence into a *well-structured* sequence.

Input

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

For each test case:

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the sequence a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the elements of the sequence.

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

Output

For each test case, output a single integer — the minimum number of elements you need to remove to make the sequence *well-structured*.

Examples

input Copy

```
7
7
3 3 4 5 2 6 1
4
5 6 3 2
6
3 4 1 6 7 7
3
1 4 3
5
1 2 3 4 5
5
1 2 3 1 2
5
4 5 5 1 5
```

output Copy

```
0
4
1
1
3
1
0
```

Note

In the first example, the sequence is already well-structured, so no deletions are needed.

In the second example, there is no way to form a well-structured sequence except by removing all elements.

In the fifth example, by removing the first and last elements, the remaining sequence $[2, 3, 4]$ becomes well-structured.

E. Secret Message

time limit per test: 1 s.
memory limit per test: 256 MB

In Prime Academy, there was a developer who loved collecting English letters. One day, he decided to write a secret message to his friend, but there was a problem — he didn't know all the letters. He only knew a few letters of the alphabet in order, starting from **a**.

The developer wondered: "How many of the first letters of the alphabet do I need to learn so that I can write my entire message?"

For example:

- If the message is `"moon"`, he needs to know up to the letter **o**, so the alphabet size = 15.
- If the message is `"zebra"`, he needs to know up to the letter **z**, so the alphabet size = 26.

Help our developer to find the **minimum alphabet size** he needs so that he can write his message without missing any letters.

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 100$) — the length of the message.

The second line of each test case contains a string s of length n, consisting of lowercase English letters — the message that the developer wants to write.

Output

For each test case, output a single integer — the minimum number of letters from the beginning of the English alphabet (starting with 'a') that the developer needs to know in order to write the entire message s.

Example

input Copy

```
5
1
a
4
down
10
codeforces
3
bcf
5
zzzzz
```

output Copy

```
1
23
19
6
26
```

Note

In the first test case, the message is `"a"`, so the developer only needs to know the first letter of the alphabet. Minimum alphabet size = 1.

In the second test case, the message is `"down"`, and the developer needs to know the letters `d`, `o`, `w`, `n`. The largest letter alphabetically is `w` (23rd letter), so the minimum alphabet size = 23.

F. Solve it if you Can !

time limit per test: 3 s.
memory limit per test: 1024 MB

The world of the black cat is collapsing.

In this world, which can be represented as a rooted tree with root at node 1, Liki and Sasami need to uncover the truth about the world.

Each day, they can explore a node u that has not yet collapsed. After this exploration, the black cat causes u and all nodes in its subtree to collapse. Additionally, at the end of the i th day, if it exists, the number $n - i + 1$ node will also collapse.

For each i from 1 to n, determine the number of exploration schemes where Liki and Sasami explore exactly i days (i.e., they perform exactly i operations), with the last exploration being at node 1. The result should be computed modulo $998\,244\,353$.

Note: It is guaranteed that nodes 1 to n can form a "DFS" order of the tree, meaning there exists a depth-first search traversal where the i th visited node is i.

Input

The first line contains an integer t ($1 \leq t \leq 10$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains exactly one number n ($3 \leq n \leq 80$).

Each of the following $n - 1$ lines contains two integers u_i and v_i, representing two vertices connected by an edge ($1 \leq u_i, v_i \leq n$). It is guaranteed that the given edges form a tree. It is also guaranteed that the vertices can form a "DFS" traversal order.

It is guaranteed that the sum of n for all test cases does not exceed 80

Output

For each test case, print n integers, where the i th integer represents the number of exploration schemes for exactly i days, modulo $998\,244\,353$.

Example

input Copy

```
2
4
1 2
2 3
2 4
7
4 2
6 1
5 1
7 6
2 3
1 2
```

output Copy

```
1 3 3 1
1 6 23 48 43 17 1
```

Note

For the first test case, the following operation sequences are legal:

$\{1\}, \{2, 1\}, \{3, 1\}, \{4, 1\}, \{3, 2, 1\}, \{4, 2, 1\}, \{4, 3, 1\}, \{4, 3, 2, 1\}$.

G. Books in Large Library

time limit per test: 1 s.
memory limit per test: 256 MB

In a large library, there are three types of books available:

- a copies of math books,
- b copies of literature books,
- c copies of encyclopedias (these can be used instead of either math or literature books).

The student Maha needs:

- x math books,
- y literature books.

Rules:

- A math requirement can be satisfied by either a math book or an encyclopedia.
- A literature requirement can be satisfied by either a literature book or an encyclopedia.
- A single book cannot be shared between two subjects or two students.

Task: Determine whether Maha can get enough books for her needs from the library.

Input

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

Then t lines are given, each containing a description of one test case. Each description consists of five integers a, b, c, x and y ($0 \le a, b, c, x, y \le 10^8$).

Output

For each test case in a separate line, output:

- `YES`, if Maha can get all the books she needs;
- `NO` otherwise.

You can output `YES` and `NO` in any case (for example, strings `yEs`, `yes`, `Yes` and `YES` will be recognized as a positive response).

Example

input Copy

```
7
1 1 4 2 3
0 0 0 0 0
5 5 0 4 6
1 1 1 1 1
50000000 50000000 100000000 100000000 100000000
0 0 0 100000000 100000000
1 3 2 2 5
```

output Copy

```
YES
YES
NO
YES
YES
NO
NO
```

H. Interactive

time limit per test: 1 s.
memory limit per test: 256 MB

This is an interactive problem. In the output section below you will see the information about flushing the output.

Ameen thinks of some hidden number — an integer from interval $[2, 100]$. Your task is to say if the hidden number is prime or composite.

Integer $x > 1$ is called prime if it has exactly two distinct divisors, 1 and x. If integer $x > 1$ is not prime, it's called composite.

You can ask up to 20 queries about divisors of the hidden number. In each query you should print an integer from interval $[2, 100]$. The system will answer `yes` if your integer is a divisor of the hidden number. Otherwise, the answer will be `no`.

For example, if the hidden number is 14 then the system will answer "`yes`" only if you print 2, 7 or 14.

When you are done asking queries, print `prime` or `composite` and terminate your program.

You will get the `Wrong Answer` verdict if you ask more than 20 queries, or if you print an integer not from the range $[2, 100]$. Also, you will get the `Wrong Answer` verdict if the printed answer isn't correct.

You will get the `Idleness Limit Exceeded` verdict if you don't print anything (but you should) or if you forget about flushing the output (more info below).

Input

After each query you should read one string from the input. It will be `yes` if the printed integer is a divisor of the hidden number, and `no` otherwise.

Output

Up to 20 times you can ask a query — print an integer from interval $[2, 100]$ in one line. You have to both print the end-of-line character and `flush` the output. After flushing you should read a response from the input.

In any moment you can print the answer `prime` or `composite` (without the quotes). After that, flush the output and terminate your program.

To flush you can use (just after printing an integer and end-of-line):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- See the documentation for other languages.

Examples

input Copy

```
yes
no
yes
```

output Copy

```
2
80
5
composite
```

input Copy

```
no
yes
no
no
no
```

output Copy

```
58
59
78
78
2
prime
```

Note

The hidden number in the first query is 30. The hidden number is divisible by both 2 and 5. Thus, it must be composite. Note that it isn't necessary to know the exact value of the hidden number. In this test, the hidden number is 30.

59 is a divisor of the hidden number. In the interval $[2, 100]$ there is only one number with this divisor. The hidden number must be 59, which is prime. Note that the answer is known even after the second query and you could print it then and terminate. Though, it isn't forbidden to ask unnecessary queries (unless you exceed the limit of 20 queries).

I. Ameen And Integers

time limit per test: 0.5 s.
memory limit per test: 256 MB

One day, Ameen decided to write all the integers from 1 to n on a sheet of paper. After finishing, he noticed the paper looked too crowded, so he decided to do something strange: he replaced each number with the sum of its digits.

For example, if the number is 12, it becomes $1 + 2 = 3$. If the number is 7, it stays 7.

After replacing all the numbers, Ameen wondered: "What is the sum of all the numbers on the paper now?"

Task Given n, find the sum of the numbers after each number has been replaced by the sum of its digits.

Input

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

The only line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the largest number Ameen wrote on the paper.

Output

For each test case, output a single integer — the sum of the numbers on the paper after the replacement process.

Example

input Copy

```
7
12
1

2
3
1434
2024
200000
```

output Copy

```
51
1
3
6
18465
28170
4600002
```

J. Tree Free

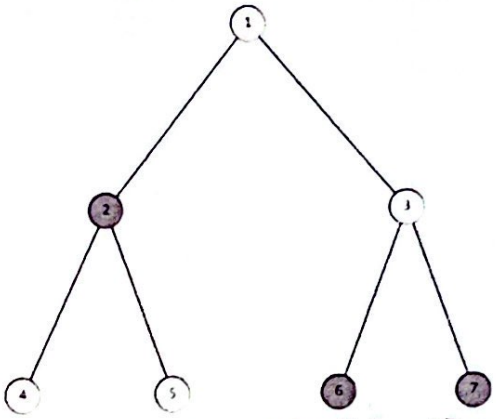
time limit per test: 2 s.

memory limit per test: 256 MB

You have a tree with n vertices, some of which are marked. A tree is a connected undirected graph without cycles.

Let f_i denote the maximum distance from vertex i to any of the marked vertices.

Your task is to find the minimum value of f_i among all vertices.



For example, in the tree shown in the example, vertices 2, 6, and 7 are marked. Then the array $f(i) = [2, 3, 2, 4, 4, 3, 3]$. The minimum f_i is for vertices 1 and 3.

Input

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

The first line of each test case contains two integers n and k ($1 \le k \le n \le 2 \cdot 10^5$) — the number of vertices in the tree and the number of marked vertices, respectively.

The second line of each test case contains k integers a_i ($1 \le a_i \le n, a_{i-1} < a_i$) — the indices of the marked vertices.

The next $n - 1$ lines contain two integers u_i and v_i — the indices of vertices connected by the i-th edge.

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

Output

For each test case, output a single integer — the minimum value of f_i among all vertices.

Examples

input

```
6
7 3
2 6 7
1 2
1 3
2 4
2 5
3 6
3 7
4 4
1 2 3 4
1 2
2 3
3 4
5 1
1
1 2
1 3
1 4
1 5
5 2
4 5
1 2
2 3
1 4
4 5
10 8
1 2 3 4 5 8 9 10
2 10
10 5
5 3
3 1
1 7
7 4
4 9
8 9
6 1
10 9
1 2 4 5 6 7 8 9 10

1 3
3 9
9 4
4 10
10 6
6 7
7 2
2 5
5 8
```

output

```
2
2
0
1
4
5
```

input

Copy

```
3
6 1
3
1 2
1 3
3 4
3 5
2 6
5 3
1 2 5
1 2
1 3
2 4
3 5
7 1
2
3 2
2 6
6 1
5 6
7 6
4 5
```

output

Copy

```
0
2
0
```

K. Identical Tiles

time limit per test: 1 s
memory limit per test: 256 MB

A group of friends are collecting identical 1×1 tiles. They have n boxes, and the i-th box contains a_i tiles.

They want to use **all** the tiles they have to build a perfect square floor (same length and width, with no gaps and no leftover tiles).

Can they do it?

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 a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of boxes.

The second line of each test case contains n integers $a_1, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the number of tiles in each box.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output "YES" if it is possible to use all the tiles to build a perfect square floor, and "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Example

input Copy

```
5
1
9
2
14 2
7
1 2 3 4 5 6 7
6
1 3 5 7 9 11
4
2 2 2 2
```

output Copy

```
YES
YES
NO
YES
NO
```

Note

- In the first test case, the friends can build a 3×3 floor.
- In the second test case, the friends can build a 4×4 floor.
- In the third test case, the friends cannot build a perfect square floor using all the given tiles.

L. A Or B

time limit per test: 1 s.
memory limit per test: 256 MB

Primo has a string of length 5, whose characters are each either A or B.

Which letter appears most frequently: A or B?

Input

The first line of the input contains an integer t ($1 \leq t \leq 32$) — the number of test cases.

The only line of each test case contains a string of length 5 consisting of letters A and B.

All t strings in a test are different (distinct)

Output

For each test case, output one letter (A or B) denoting the character that appears most frequently in the string.

Example

input (Copy)

```
8
ABABB
ABABA
BBBAB
AAAAA
BBBBB
BABAA
AAAAB
BAAAA
```

output (Copy)

```
B
A
B
A
B
A
A
A
```

THE END...