A. Shuffle Hashing

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

Polycarp has built his own web service. Being a modern web service it includes login feature. And that always implies password security problems.

Polycarp decided to store the hash of the password, generated by the following algorithm:

1. take the password p, consisting of lowercase Latin letters, and shuffle the letters randomly in it to obtain p' (p' can still be equal to p);
2. generate two random strings, consisting of lowercase Latin letters, s_1 and s_2 (any of these strings can be empty);
3. the resulting hash $h = s_1 + p' + s_2$, where addition is string concatenation.

For example, let the password $p =$ "abacaba". Then p' can be equal to "aabcaab". Random strings $s1 =$ "zyx" and $s2 =$ "kjh". Then $h =$ "zyxaabcaabkjh".

Note that no letters could be deleted or added to p to obtain p', only the order could be changed.

Now Polycarp asks you to help him to implement the password check module. Given the password p and the hash h, check that h can be the hash for the password p.

Your program should answer t independent test cases.

Input

The first line contains one integer t ($1 \le t \le 100$) — the number of test cases.

The first line of each test case contains a non-empty string p, consisting of lowercase Latin letters. The length of p does not exceed 100.

The second line of each test case contains a non-empty string h, consisting of lowercase Latin letters. The length of h does not exceed 100.

Output

For each test case print the answer to it — "YES" if the given hash h could be obtained from the given password p or "NO" otherwise.

Example

input

```
5
abacaba
zyxaabcaabkjh
onetwothree
threetwoone
one
zzonneyy
one
none
twenty
ten
```

output

```
YES
YES
NO
YES
NO
```

Note

The first test case is explained in the statement.

In the second test case both s_1 and s_2 are empty and $p' =$ "threetwoone" is p shuffled.

In the third test case the hash could not be obtained from the password.

In the fourth test case $s_1 =$ "n", s_2 is empty and $p' =$ "one" is p shuffled (even thought it stayed the same).

In the fifth test case the hash could not be obtained from the password.

B. A and B

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two integers a and b. You can perform a sequence of operations: during the first operation you choose one of these numbers and increase it by 1; during the second operation you choose one of these numbers and increase it by 2, and so on. You choose the number of these operations yourself.

For example, if $a = 1$ and $b = 3$, you can perform the following sequence of three operations:

1. add 1 to a, then $a = 2$ and $b = 3$;
2. add 2 to b, then $a = 2$ and $b = 5$;
3. add 3 to a, then $a = 5$ and $b = 5$.

Calculate the minimum number of operations required to make a and b equal.

Input

The first line contains one integer t ($1 \le t \le 100$) — the number of test cases.

The only line of each test case contains two integers a and b ($1 \le a, b \le 10^9$).

Output

For each test case print one integer — the minimum numbers of operations required to make a and b equal.

Example

input

```
3
1 3
11 11
30 20
```

output
3 0 4

Note

First test case considered in the statement.

In the second test case integers a and b are already equal, so you don't need to perform any operations.

In the third test case you have to apply the first, the second, the third and the fourth operation to b (b turns into $20 + 1 + 2 + 3 + 4 = 30$).

C. Berry Jam

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

Karlsson has recently discovered a huge stock of berry jam jars in the basement of the house. More specifically, there were $2n$ jars of strawberry and blueberry jam.

All the $2n$ jars are arranged in a row. The stairs to the basement are exactly in the middle of that row. So when Karlsson enters the basement, he sees exactly n jars to his left and n jars to his right.

For example, the basement might look like this:

Being the starightforward man he is, he immediately starts eating the jam. In one minute he chooses to empty either the first non-empty jar to his left or the first non-empty jar to his right.

Finally, Karlsson decided that at the end the amount of full strawberry and blueberry jam jars should become the same.

For example, this might be the result:

He has eaten 1 jar to his left and then 5 jars to his right. There remained exactly 3 full jars of both strawberry and blueberry jam.

Jars are numbered from 1 to $2n$ from left to right, so Karlsson initially stands between jars n and $n+1$.

What is the minimum number of jars Karlsson is required to empty so that an equal number of full strawberry and blueberry jam jars is left?

Your program should answer t independent test cases.

Input

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

The first line of each test case contains a single integer n ($1 \le n \le 10^5$).

The second line of each test case contains $2n$ integers $a_1, a_2, \dots, a_{2n}$ ($1 \le a_i \le 2$) — $a_i = 1$ means that the i-th jar from the left is a strawberry jam jar and $a_i = 2$ means that it is a blueberry jam jar.

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

Output

For each test case print the answer to it — the minimum number of jars Karlsson is required to empty so that an equal number of full strawberry and blueberry jam jars is left.

Example

input
4 6 1 1 1 2 2 1 2 1 2 1 1 2 2 1 2 1 2 3 1 1 1 1 1 1 2 2 1 1 1
output
6 0 6 2

Note

The picture from the statement describes the first test case.

In the second test case the number of strawberry and blueberry jam jars is already equal.

In the third test case Karlsson is required to eat all 6 jars so that there remain 0 jars of both jams.

In the fourth test case Karlsson can empty either the second and the third jars or the third and the fourth one. The both scenarios will leave 1 jar of both jams.

D. Segment Tree

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

As the name of the task implies, you are asked to do some work with segments and trees.

Recall that a tree is a connected undirected graph such that there is exactly one simple path between every pair of its vertices.

You are given n segments $[l_1, r_1], [l_2, r_2], \dots, [l_n, r_n]$, $l_i < r_i$ for every i. It is guaranteed that all segments' endpoints are integers, and all endpoints are unique — there is no pair of segments such that they start in the same point, end in the same point or one starts in the same point the other one ends.

Let's generate a graph with n vertices from these segments. Vertices v and u are connected by an edge if and only if segments $[l_v, r_v]$ and $[l_u, r_u]$ intersect and neither of it lies fully inside the other one.

For example, pairs $([1,3],[2,4])$ and $([5,10],[3,7])$ will induce the edges but pairs $([1,2],[3,4])$ and $([5,7],[3,10])$ will not.

Determine if the resulting graph is a tree or not.

Input

The first line contains a single integer n ($1 \le n \le 5 \cdot 10^5$) — the number of segments.

The i-th of the next n lines contain the description of the i-th segment — two integers l_i and r_i ($1 \le l_i < r_i \le 2n$).

It is guaranteed that all segments borders are pairwise distinct.

Output

Print "YES" if the resulting graph is a tree and "NO" otherwise.

Examples

input

```
6
9 12
2 11
1 3
6 10
5 7
4 8
```

output

```
YES
```

input

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

output

```
NO
```

input

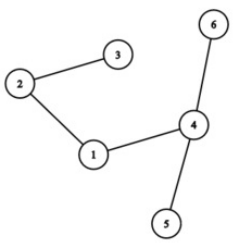
```
5
5 8
3 6
2 9
7 10
1 4
```

output

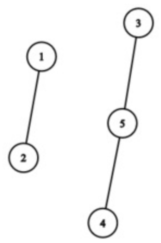
```
NO
```

Note

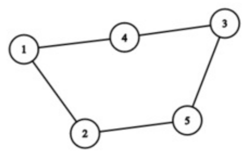
The graph corresponding to the first example:



The graph corresponding to the second example:



The graph corresponding to the third example:



E. Tests for problem D

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

We had a really tough time generating tests for problem D. In order to prepare strong tests, we had to solve the following problem.

Given an undirected labeled tree consisting of n vertices, find a set of segments such that:

1. both endpoints of each segment are integers from 1 to $2n$, and each integer from 1 to $2n$ should appear as an endpoint of exactly one segment;
2. all segments are non-degenerate;
3. for each pair (i, j) such that $i \neq j$, $i \in [1, n]$ and $j \in [1, n]$, the vertices i and j are connected with an edge if and only if the segments i and j intersect, but neither segment i is fully contained in segment j, nor segment j is fully contained in segment i.

Can you solve this problem too?

Input

The first line contains one integer n ($1 \le n \le 5 \cdot 10^5$) — the number of vertices in the tree.

Then $n - 1$ lines follow, each containing two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \neq y_i$) denoting the endpoints of the i-th edge.

It is guaranteed that the given graph is a tree.

Output

Print n pairs of integers, the i-th pair should contain two integers l_i and r_i ($1 \le l_i < r_i \le 2n$) — the endpoints of the i-th segment. All $2n$ integers you print should be unique.

It is guaranteed that the answer always exists.

Examples

input
```
6
1 2
1 3
3 4
3 5
2 6
```
output
```
9 12
7 10
3 11
1 5
2 4
6 8
```

input
```
1
```
output
```
1 2
```

F. Cards

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Consider the following experiment. You have a deck of m cards, and exactly one card is a joker. n times, you do the following: shuffle the deck, take the top card of the deck, look at it and return it into the deck.

Let x be the number of times you have taken the joker out of the deck during this experiment. Assuming that every time you shuffle the deck, all $m!$ possible permutations of cards are equiprobable, what is the expected value of x^k? Print the answer modulo 998244353.

Input

The only line contains three integers n, m and k ($1 \le n, m < 998244353$, $1 \le k \le 5000$).

Output

Print one integer — the expected value of x^k, taken modulo 998244353 (the answer can always be represented as an irreducible fraction $\frac{a}{b}$, where $b \mod 998244353 \neq 0$; you have to print $a \cdot b^{-1} \mod 998244353$).

Examples

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

input
```
1 1 5000
```
output
```
1
```

input
```
2 2 2
```
output
```
499122178
```

input
```
998244352 1337 5000
```
output
```
326459680
```

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform