A. Bookshelf Filling

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

Recently, Joseph bought lots of books and was going to put them all on the shelf. He found that there are only two types of books – the type A with height a and thickness 1, and the type B with height b $(a < b)$ and thickness 1. The number of each type of book are n and m respectively. And he already knew the height of the bookshelf which is h $(h \geq b)$.

In the beginning, he arranged all the books on the shelf in height-ascending order, which means the books of height a are all on the left, and the books of height b are all on the right. The total length these books occupied on the bookshelf will be $n + m$.

However, Joseph found there was still a vacancy on top of these books, so he prepared to select k $(0 \leq k \leq m - 1)$ books with height b from the **rightmost** side and put them all horizontally on top of other books to reduce the length they occupied. He wanted to know the minimum width if he arranges these books optimally.

Input

The first line of the input gives the number of test cases T $(1 \leq T \leq 10^3)$. T test cases follow.

For each test case, only one line contains five integers a, b, n, m, h $(1 \leq a < b \leq h \leq 10^6, 1 \leq n, m \leq 10^9)$

Output

For each test case, print a line contains one integer w, indicating the minimum width.

Example

input

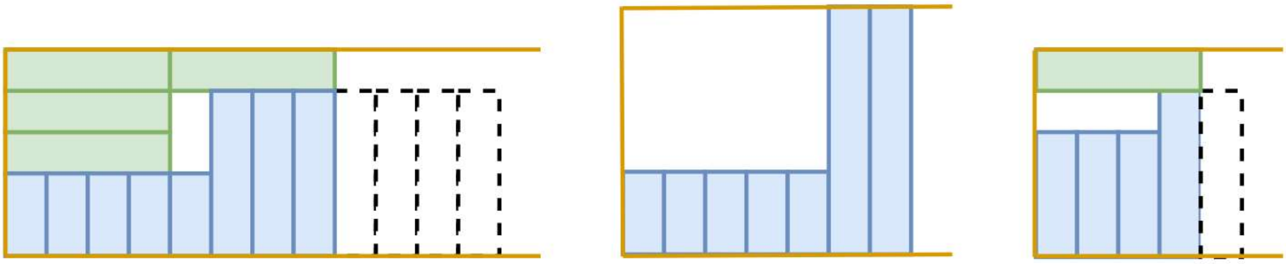
```
4
2 5 5 8 5
2 4 5 7 5
2 6 5 2 6
3 4 3 2 5
```

output

```
10
8
7
4
```

Note

The following pictures from left to right show the possible arrangement of books for the testcases $2, 3$ and 4.



B. Kanbun

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Paulliant is a linguist. Recently he received an article written in language A and he was told to translate it into language B. The two languages only differ in word order, so, Paulliant decided to translate by redefining the reading order of the articles, and came up with the following method.

Suppose that the article consists of n words, numbered as $1, 2, 3 \cdots n$. Paulliant generated a string str consisting of only '(', ')' and '-', whose length is also n. The string must satisfy the rule of bracket matching. Formally speaking, define $s_{(i}$ and $s_{)i}$, representing the number of character '(' or ')' within the first i characters of the string, respectively, there is $s_{(i} \geq s_{)i}$ for every $1 \leq i \leq n$ and $s_{(n} = s_{)n}$.

The process to read the article obeys the following rules.

(1) The process starts from the first word.

(2) For the i-th word, if str_i is '-' or ')', then read the i-th word directly, and jump to the $(i+1)$-th word.

(3) For the i-th word, if str_i is '(', then find the matching right bracket of it. Supposing it to be str_j, read the $(i+1)$-th to j-th word, then read the i-th word, and finally jump to the $(j+1)$-th word.

(4) If we are to read the $(n+1)$-th word (obviously it doesn't exist), the process ends.

Note that **the process is recursive**, meaning that when reading the $(i+1)$-th to j-th word in rule (2), it still follows the four rules.

For example,

- If $n = 4$, str is "(-)-", the reading order will be 2, 3, 1, 4.
- If $n = 7$, str is "-((-)-)", the reading order will be 1, 4, 5, 3, 6, 7, 2.

Now, give you the number of words n, and the string str Paulliant generated. Your task is to print the reading order following the four rules. It can be proved that the reading order should be **a permutation** of numbers from 1 to n.

Input

The first line contains a single integer n $(3 \leq n \leq 10^5)$ — the number of words.

The second line contains a string str of n characters. It consists of n characters of only '(', ')', '-', and it's guaranteed to satisfy the rules of bracket matching.

Output

Output a line consisting of n integers which represent the reading order, it should be a permutation of numbers from 1 to n. The n integers are separated by spaces.

Examples

input

```
4
(-)-
```

output

```
2 3 1 4
```

input

```
7
-((-)-)
```

output

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

input

```
14
((-)-()--)(-)-
```

output

```
3 4 2 5 7 6 8 9 10 1 12 13 11 14
```

C. Tree Division

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

You are given a tree of n nodes, the i-th node has value a_i.

Define a node t is valid, if you can divide the n nodes into two sets A and B, such that the following condition would hold:

- $\forall u, v \in A(u \neq v)$, if u is on the path from t to v, then $a_u < a_v$
- $\forall u, v \in B(u \neq v)$, if u is on the path from t to v, then $a_u > a_v$

You need to find out whether the node 1 is valid.

Input

The first line contains one integer $n(1 \leq n \leq 10^5)$.

The second line contains n integers $a_1, a_2, \cdots, a_n (1 \le a_i \le n)$.

The next $n-1$ lines each contains two integers x and y $(1 \le x, y \le n)$, denoting an edge between node x and y.

Output

Output `YES` if node 1 is valid, otherwise print `NO`.

Examples

input

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

output

```
YES
```

input

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

output

```
NO
```

Note

In this case, node 1 is valid.

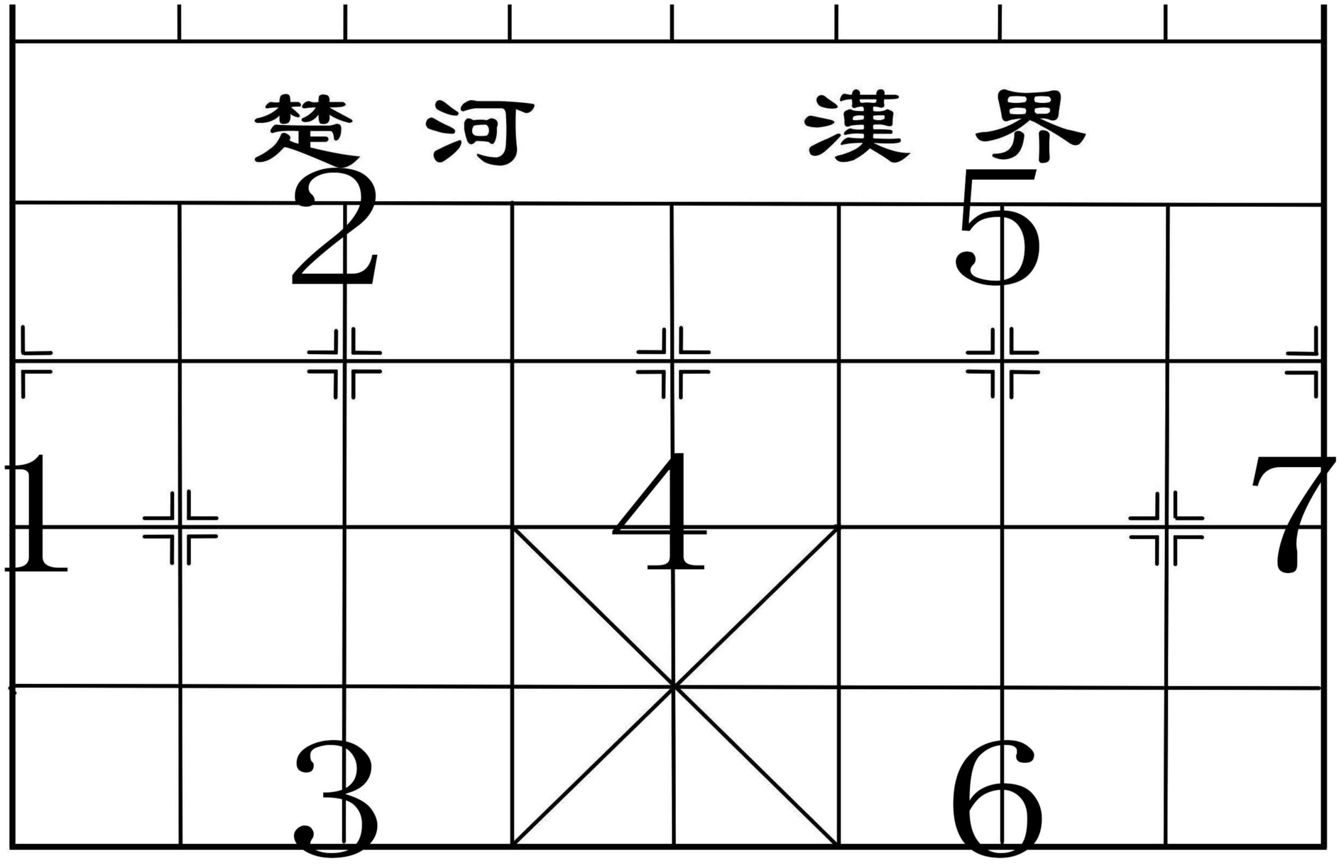
A possible division is: $A = \{2, 3\}$,$B = \{1, 4, 5, 6, 7, 8\}$.

D. 342 and Xiangqi

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

342 likes playing Xiangqi (as it's called Chinese chess). His favorite piece in Xiangqi is "Xiang" because his opponents usually overlook the existence of Xiang so their major pieces are often taken by 342's Xiang.

For those of you who don't know how Xiang moves in Xiangqi, Xiang is a minor piece that each player initially has two, located at the position 3 and 6 respectively. The number for each position is showed in the illustration bellow. In a single move, Xiang should move 2 squares horizontally as well as vertically. But note that it can neither move out of the half board nor it can overlap with other pieces. The illustration below showed all the 7 possible positions that Xiang could reach. For example, in a single move,



Centered scaled image.

- A Xiang at position 2 **can** move to position 4,
- A Xiang at position 7 **can** move to position 6,
- A Xiang at position 3 **cannot** move to position 5.

342 has 2 Xiangs as the only 2 pieces in the half board. Their present positions are given and guaranteed to be 2 different positions out of the 7 possible ones. But 342 disliked the present positions of his 2 Xiangs and would like to adjust their positions by a few moves. In each move, he can choose one of the Xiangs and move it following the rules explained. The final positions 342 want are also given and guaranteed to be 2 different positions out of the 7 possible ones. Note that **the 2 Xiangs are considered the same**, meaning that he doesn't restrict that which of the 2 Xiangs has to move to which of the 2 positions.

Now 342 wonders, **at least how many moves** he need to adjust the 2 Xiangs from the present positions to the final positions that 342 want?

Input

The first line contains a single integer T $(1 \le T \le 10^5)$ – the number of test cases.

This is followed by T lines, representing each test case, each containing a single line of four integers a_1, a_2, b_1, b_2 such that $1 \le a_1, a_2, b_1, b_2 \le 7,\ a_1 \ne a_2,\ b_1 \ne b_2$. a_1, a_2 represent the present positions of the 2 Xiangs while b_1, b_2 represent the final positions of them.

Output

For each test case, print a single integer: the least number of moves that 342 needs to adjust his 2 Xiangs to the final positions, considering the 2 Xiangs to be the same?

Example

input

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

output

```
1
2
0
```

E. Chevonne's Necklace

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

For her excellent performance, pianist Chevonne was awarded a pearl necklace. The necklace consisted of n pearls which form a circle and are numbered $1, 2, \cdots, n$ clockwise.

To prevent the pearls from being covered with dust, she wants to move pearls from the necklace and wipe them. However, the way to remove the pearls is complex because of their uniqueness of them.

Specifically, every pearl on the necklace has a continuity value c_i $(c_i \geq 0)$. Each turn, Chevonne can choose a pearl i with $c_i \geq 1$ and move the pearl i and $c_i - 1$ clockwise consecutive pearls (c_i pearls in all). Be careful! If the number of remaining pearls is **less than** c_i, she will be unable to choose pearl i. After that, the remaining pearls will form a new circle.

Chevonne will repeat the process till she can't remove any pearls or all the pearls have been removed. She is curious about the maximum number of pearls she could remove and the number of such solutions.

Here, we use a set to describe a solution. The set contains the number i if Chevonne chose pearl i in some step. Two solutions are different if and only if the sets are different.

Input

The first line of input contains one positive integer n $(1 \leq n \leq 2000)$.

The following one line contains n non-negative integers separated by spaces. The i-th number represents c_i $(0 \leq c_i \leq 2000)$. Pearls are numbered clockwise from 1 to n.

Output

Please output two integers separated by one space.

The first one is the maximum number of pearls Chevonne could remove and the second one is the number of such solutions. Since the number of solutions may be very large, please output it modular 998244353.

Example

input

```
6
0 1 1 3 3 1
```

output

```
6 3
```

Note

In the example, all the pearls could be removed by Chevonne, the 3 possible solutions are shown as follows:

1. remove the $2, 3, 6, 4$-th pearl in order.
2. remove the $2, 3, 6, 5$-th pearl in order.
3. remove the $5, 4$-th pearl in order.

F. Let's Swap

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

Joseph developed an editing software called Pandote recently, and now he is testing it to make sure it works correctly, otherwise, he will probably get fired!

Joseph starts his testing by implementing copying and pasting as well as reversing operations to the string on the Pandote. More specifically, in each step, if the string on the screen is S, he will do the following operations in order.

1. Choose a prefix of length $l\ (1 \le l \le |S|)$, then S can be denoted by $AB\ (|A| = l)$. Note that the string B can be empty.
2. Swap the two parts and get the string BA.
3. Reverse the whole string and get the string $(BA)^r$

However, since the function of Pandote is limited, there are only two different lengths l_1 and l_2 of prefix he can choose in each step. Now Joseph wants to know whether he can convert the string S to T through several (possibly zero) steps.

Input

The first line of the input gives the number of test cases $T\ (1 \le T \le 5 \times 10^5)$. T test cases follow.

For each test case, the first line contains the string S and the second line contains the string T. Both S and T are consisting of lowercase Latin letters and $|S| = |T|$.

the third line contains two integers l_1 and l_2 $(1 \le l_1, l_2 \le |S|, l_1 \ne l_2)$, denoting the length of the prefix he can choose in each step.

It is guaranteed that the sum of $|S|$ over all test cases does not exceed 5×10^5.

Output

For each test case, print `yes` if he can convert S to T. Otherwise, print `no`.

Example

input

```
3
ljhelloh
hellohlj
2 4
thisisastr
htrtsasisi
3 5
abcde
bcdea
1 4
```

output

```
yes
no
yes
```

Note

For the first testcase, one possible method is **`ljhelloh`** $\xrightarrow{4}$ **`ehjlholl`** $\xrightarrow{2}$ **`hellohlj`**

G. Equal Sum Arrays

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Define function $f(n)$ to be the number of different array a of positive integers, such that the sum of the elements in array a equals to n.

For example, $f(3) = 4$, and the four different arrays are $\{1,1,1\}, \{1,2\}, \{2,1\}, \{3\}$.

You are given two positive integers L and R, please find the answer of $\sum\limits_{k=L}^{R} f(k)$. Since the answer may be too large, output it modulo 998244353.

Input

The first line contains a single integer T $(1 \le T \le 10^5)$ — the number of test cases.

This is followed by T lines, representing each test case, each containing a single line of two integers L, R $(1 \le L \le R \le 10^5)$.

Output

For each test case, print a single integer, representing the answer for $\sum_{k=L}^{R} f(k)$ modulo 998244353.

Examples

input
2
output
2

input
3
output
4

input
4
output
8