# Problem A. Family

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

The friends of Vus the Cossack: Fedir and Ksenia — twins in a large family.

One day, during a lunch break, Vus asked the twins to tell something about their family. Fedir said that he has $b_1$ brothers and $s_1$ sisters, while Ksenia has $b_2$ brothers and $s_2$ sisters.

Vus the Cossack remembered these four numbers, but he forgot who among the twins answered first.

That is, remembering that four numbers $b_1, s_1, b_2, s_2$ were mentioned, he is not sure whether $b_1, s_1$ is Fedir's answer or Ksenia's. However, he is sure that $b_1$ and $b_2$ denote brothers, while $s_1$ and $s_2$ denote sisters.

Vus became curious about how many brothers and sisters there are in their family in total, so with these four numbers, he turned to you for help.

## Input

The first line contains four integers $b_1$, $s_1$, $b_2$, and $s_2$ ($0 \leq b_i, s_i \leq 20$; $|b_1 - b_2| = 1$; $|s_1 - s_2| = 1$) — the responses of the twins.

It is guaranteed that a solution exists.

## Output

In a single line, output two integers — the number of brothers and sisters in the family, respectively.

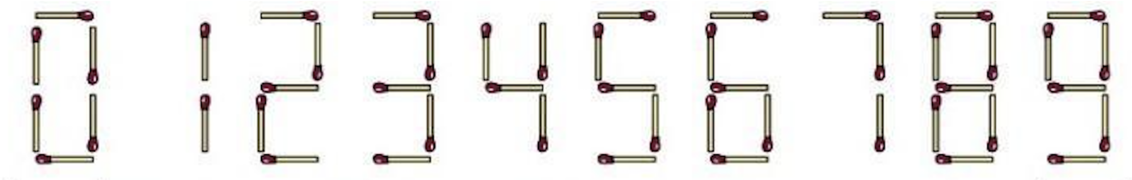
## Examples

| standard input | standard output |
|---|---|
| 1 5 2 4 | 2 5 |
| 10 6 9 7 | 10 7 |

# Problem B. A+B=C

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

Us the Cossack laid out matches on the table to form the expression (not necessarily correct) $a + b = c$, where $a, b, c$ are single-digit integers from 0 to 9.



Vus the Cossack noticed this expression and wants to move the matches **only** in the numbers $a$ and $b$ so that the expression becomes correct. That is, the `+`, `=` and $c$ cannot be changed.

Can Vus make the expression correct?

## Input

The first line contains three integers $a$, $b$, and $c$ ($0 \leq a, b, c \leq 9$).

## Output

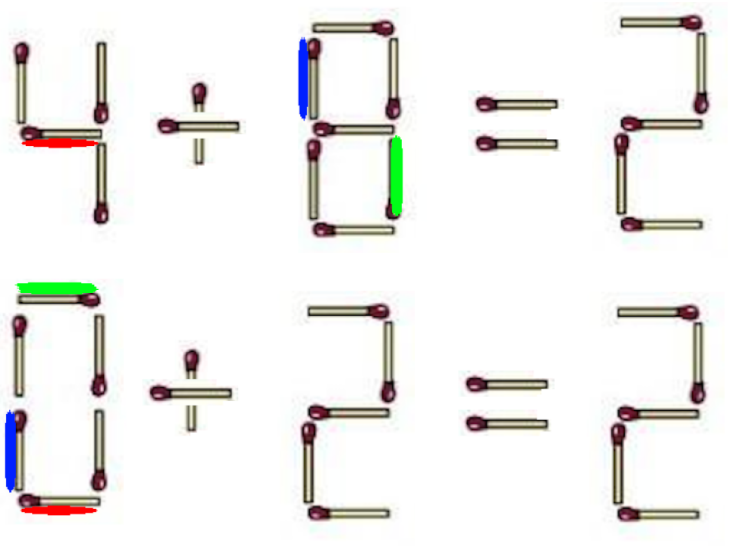
Print "`Yes`", if it is possible to do so, or "`No`" otherwise.

## Examples

| standard input | standard output |
|---|---|
| `4 8 2` | `Yes` |
| `5 0 5` | `Yes` |
| `1 5 8` | `No` |

## Note

In the first example:



In the second example, the equality holds immediately.

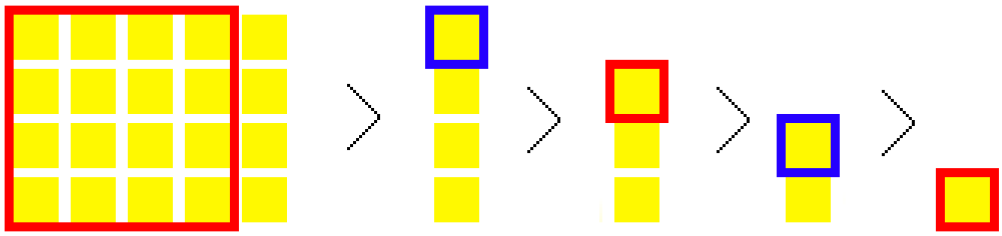
In the third example, it is impossible to rearrange the matches.

# Problem C. Area of the Cake

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

Vus the Cossack and Us the Cossack are playing a game on a cake in the shape of a rectangle with dimensions $n \times m$.

They take turns (Vus starts), cutting a square piece from the cake with the maximum possible side length such that three of the four sides of the square coincide with the sides of the cake at the beginning of the turn. The player then takes this piece for themselves. If the cake is square, the player takes the entire remaining cake.



Cutting the cake $4 \times 5$. The pieces of cake outlined in red are cut by Vus, and those in blue are cut by Us.

When the entire cake has been successfully divided, it turns out that the sum of the areas of the squares that Vus took is $p$, and Us took $q$.

The Cossacks got so caught up in the game that they forgot the size of the cake, so they asked you for help. Find any possible dimensions of the initial cake.

## Input

The first line contains two integers $p$ and $q$ ($0 \le p, q \le 10^{12}; p + q > 0$).

## Output

Output two integers $n$ and $m$ — the dimensions of the initial cake. If there are multiple correct answers, output any pair.

If such a cake does not exist, output $-1$.

## Examples

| standard input | standard output |
|---|---|
| `18 2` | `4 5` |
| `4 0` | `2 2` |
| `8 3` | `-1` |

## Note

An illustration of the first example is in the legend.

In the second example, a cake of size $2 \times 2$ satisfies the condition because then Vus will take the entire cake of area 4 on his first move.

# Problem D. Digital Game

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

Vus the Cossack and Us the Cossack are playing a game on a string $s$ of length $n$, consisting of digits 0-9.

The players take turns (Vus starts) removing any digit from the string $s$. If at any moment there are two identical digits next to each other in the string, Us wins. If all digits are removed and Us has not won, then Vus wins.

Vus the Cossack is so impatient that even before the game starts, he wants to know if he can win with optimal play (when both players always play to win) against Us, and he has asked you to find this out.

## Input

The first line contains $t$ ($1 \le t \le 10$) — the number of subtests.

In each test case:

The first line contains a single integer $n$ ($1 \le n \le 10^6$).

The second line contains a string $s$ of length $n$, consisting only of digits 0-9.

It is guaranteed that the sum of $n$ across all subtests does not exceed $10^6$.

## Output

For each of the $t$ lines, output "`Yes`" if Cossack Vus can win; or "`No`" otherwise.

## Scoring

1. (8 points): the number of different digits $\le 2$;
2. (8 points): the number of different digits $\le 5$;
3. (6 points): the number of different digits $\le 7$;
4. (8 points): only one digit appears more than once;
5. (7 points): if $s_l = s_r, s_i = s_j$ and $s_i \ne s_l$ ($l \ne r; i \ne j$), then the intervals $[l, r]$ and $[i, j]$ do not overlap;
6. (7 points): $n \le 4$;
7. (6 points): $n \le 8$;
8. (6 points): $n \le 12$;
9. (6 points): $n \le 15$;
10. (38 points): no additional restrictions.

The constraints apply to each of the $t$ subtests.

## Example

| standard input | standard output |
|---|---|
| 4<br>6<br>015423<br>7<br>1235212<br>4<br>1111<br>6<br>156156 | Yes<br>Yes<br>No<br>No |

## Note

In the first example, two identical digits will never be next to each other, as each digit appears no more than once.

In the second example, Vus can take the last 2. Then, if Us takes 1 or 2, Vus takes 2 or 1 respectively, and then all digits become different; thus, Vus will win. However, if Us takes 3 or 5, then Vus will take any 2 first, and then any 1.

In the third example, Us wins even before the game starts.

# Problem E. OldPost — NewPost

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

In the country of Trilyandia, there are $n$ cities connected by $n - 1$ roads. The road system in this country is special:

- one road connects only two different cities;
- there can be at most one road between any two cities;
- from any city, it is possible to reach any other city, possibly passing through others.

The postal company OldPost decided to provide delivery services in this country. Due to the limited number of staff, transportation is carried out only along one route $a$ of length $l$, that is, a sequence of cities $a_1, a_2, \ldots, a_l$ such that

- $a_i \neq a_j$ for $(i \neq j)$;
- cities $a_i$ and $a_{i+1}$ are connected by a road $(1 \leq i < l)$.

To maximize profit, OldPost decided to operate along the route with the maximum length.

An ancient competitor of OldPost — NewPost decided to do the same, but along its own route $b$: $b_1, b_2, \ldots, b_l$, which does not necessarily differ from $a$, but also has the maximum possible length.

The president, upon learning about the intentions of both companies, asked OldPost and NewPost to choose routes $a$ and $b$ in such a way as to cover the maximum number of cities (that is, to ensure that the maximum number of cities are in at least one route). Since the president of Trilyandia specializes in management, while the postal companies specialize in delivering items rather than finding routes, you have been tasked with finding such $a$ and $b$.

Note that each company wants to have the maximum possible length of its route; even if choosing a shorter route increases the total number of cities covered.

Help find routes $a$ and $b$ that have the maximum possible length and cover the maximum number of cities.

## Input

The first line contains a single integer $n$ $(2 \leq n \leq 6 \cdot 10^5)$ — the number of cities in the country.

Each of the next $n - 1$ lines contains two integers $x_i$ and $y_i$ $(1 \leq x_i, y_i \leq n, x_i \neq y_i)$ — pairs of cities between which there is a road.

## Output

The first line should contain a single integer $l$ $(2 \leq l \leq n)$ — the length of the found trade routes.

The second line should contain $l$ integers $a_1, a_2, \ldots, a_l$ $(1 \leq a_i \leq n)$ — the cities in route $a$.

The third line should contain $l$ integers $b_1, b_2, \ldots, b_l$ $(1 \leq b_i \leq n)$ — the cities in route $b$.

If there are multiple optimal routes, output any of them.

## Scoring

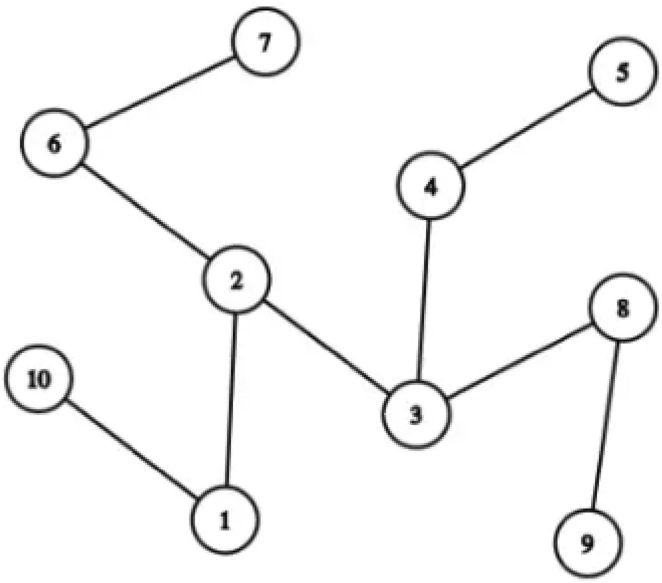
1. (8 points): $n \leq 10$;

2. (8 points): $n \le 20$;
3. (6 points): $n \le 1\,000$; there is exactly one city that is directly connected to all others;
4. (8 points): $n \le 20\,000$; there are exactly two cities that are connected to only one other city (the cities form a line);
5. (8 points): $n \le 20\,000$; there is exactly one city that is connected to three or more cities;
6. (8 points): $n \le 10\,000$; there are no more than 10 cities that are connected to only one other city;
7. (16 points): $n \le 1\,000$;
8. (18 points): $n \le 70\,000$;
9. (20 points): no additional restrictions.

## Example

| standard input | standard output |
|---|---|
| 10<br>1 2<br>2 3<br>3 4<br>4 5<br>2 6<br>6 7<br>3 8<br>8 9<br>10 1 | 6<br>5 4 3 2 1 10<br>9 8 3 2 6 7 |

## Note




In the example, the longest possible routes are as follows:

1. 5-4-3-2-6-7
2. 9-8-3-2-6-7
3. 5-4-3-2-1-10
4. 9-8-3-2-1-10

The first and fourth routes allow covering all cities.