

## Problem A. Család

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         256 megabyte

Vus, a kozák barátai: Fedir és Ksenia — ikrek egy nagy családban.

Egy nap, az ebédszünet alatt, Vus megkérte az ikreket, hogy meséljenek valamit a családjukról. Fedir azt mondta, hogy neki  $b_1$  fiútestvére és  $s_1$  lánytestvére van, míg Ksenianak  $b_2$  fiútestvére és  $s_2$  lánytestvére van.

Vus, a kozák, emlékezett ezekre a négy számra, de elfelejtette, hogy az ikrek közül ki válaszolt először.

Ezért, emlékezve arra, hogy a négy szám  $b_1, s_1, b_2, s_2$  elhangzott, nem biztos abban, hogy  $b_1, s_1$  Fedir válasza vagy Kseniaé. Azonban biztos abban, hogy  $b_1$  és  $b_2$  a fiútestvéreket, míg  $s_1$  és  $s_2$  a lánytestvéreket jelöli.

Vus kíváncsi lett arra, hogy összesen hány fiú- és lánytestvér van a családjukban, ezért ezekkel a négy számmal hozzád fordult segítségért.

### Input

Az első sor négy egész számot tartalmaz:  $b_1, s_1, b_2$  és  $s_2$  ( $0 \leq b_i, s_i \leq 20$ ;  $|b_1 - b_2| = 1$ ;  $|s_1 - s_2| = 1$ ) — az ikrek válaszai.

Garantált, hogy létezik megoldás.

### Output

Egy sorban, írj ki két egész számot — a családban található fiú- és lánytestvérek számát, sorrendben.

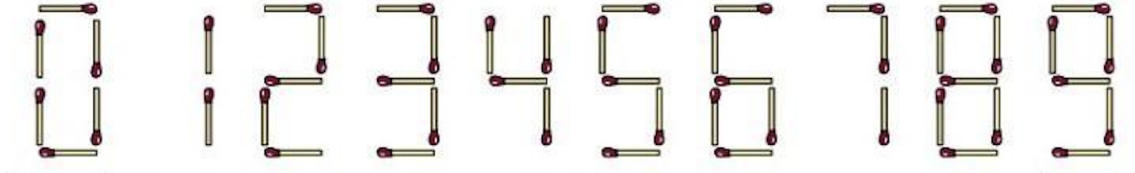
### Examples

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

Vus a kozák gyufákat rakott ki az asztalon, hogy létrehozza a kifejezést (nem feltétlenül helyes)  $a + b = c$ , ahol  $a, b, c$  egyjegyű egész számok 0 és 9 között.



Vus a kozák észrevette ezt a kifejezést, és szeretné a gyufákat **csak** az  $a$  és  $b$  számokban elmozdítani, hogy a kifejezés helyessé váljon. Tehát a  $+$ ,  $=$  és  $c$  nem változtatható.

Meg tudja Vus csinálni, hogy a kifejezés helyes legyen?

### Input

Az első sorban három egész szám  $a$ ,  $b$  és  $c$  ( $0 \leq a, b, c \leq 9$ ) található.

### Output

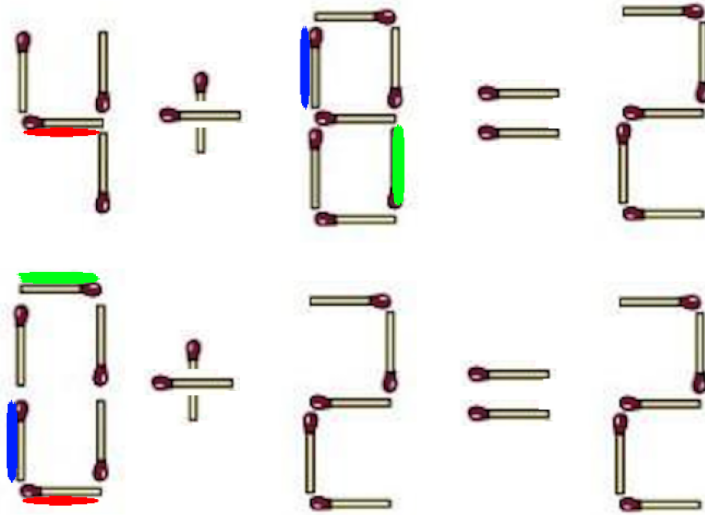
Írja ki "Igen", ha lehetséges, vagy "Nem", ha nem.

### Examples

standard input	standard output
4 8 2	Yes
5 0 5	Yes
1 5 8	No

### Note

Az első példában:



A második példában az egyenlőség azonnal teljesül.

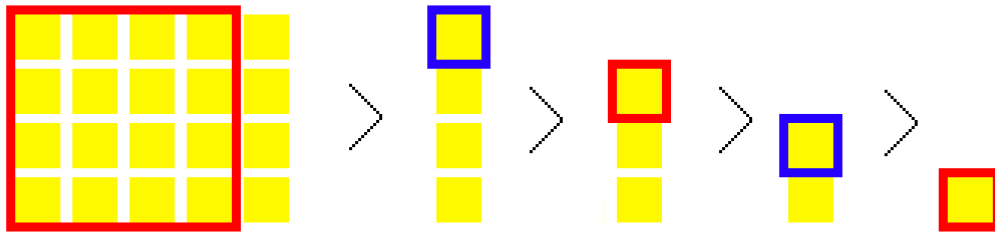
A harmadik példában lehetetlen átrendezni a gyufákat.

## Problem C. A Torta Területe

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

Vus a kozák és Us a kozák egy játékot játszanak egy téglalap alakú tortán, amelynek méretei  $n \times m$ .

Felváltva lépnek (Vus kezd), és a tortából a lehető legnagyobb oldalhosszúságú négyzetet vágják ki úgy, hogy a négyzet négy oldalának három oldala a torta oldalával egybeessen a lépés elején. A játékos ezután magához veszi ezt a darabot. Ha a torta négyzet alakú, a játékos az egész megmaradt tortát elveszi.



A torta vágása  $4 \times 5$ . A pirossal körvonalazott tortadarabokat Vus vágta ki, a kékkel körvonalazottakat pedig Us.

Amikor az egész tortát sikeresen felosztották, kiderül, hogy a Vus által elvett négyzetek területeinek összege  $p$ , míg Us által elvett négyzetek területe  $q$ .

A kozákok annyira belemerültek a játékba, hogy elfelejtették a torta méretét, ezért segítséget kértek tőled. Találd meg a kezdeti torta bármilyen lehetséges méreteit.

### Input

Az első sorban két egész szám  $p$  és  $q$  ( $0 \leq p, q \leq 10^{12}; p + q > 0$ ).

### Output

Írd ki a két egész számot  $n$  és  $m$  – a kezdeti torta méretei. Ha több helyes válasz is létezik, írd ki bármely párt.

Ha ilyen torta nem létezik, írd ki  $-1$ .

### Examples

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

### Note

Az első példa illusztrációja a legendában található.

A második példában egy  $2 \times 2$  méretű torta megfelel a feltételnek, mert akkor Vus az első lépésében az egész tortát elveszi, amelynek területe 4.

## Problem D. Digitális Játék

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

Vus a kozák és Us a kozák egy  $s$  hosszúságú,  $n$ -ből álló játékon játszik, amely csak 0-9 közötti számjegyekből áll.

A játékosok felváltva (Vus kezd) eltávolítanak bármelyik számjegyet a  $s$  karakterláncból. Ha bármikor két azonos számjegy egymás mellett van a karakterláncban, Us nyer. Ha az összes számjegy eltávolításra kerül, és Us nem nyert, akkor Vus nyer.

Vus a kozák annyira türelmetlen, hogy még a játék kezdete előtt tudni akarja, hogy nyerhet-e optimális játék mellett (amikor mindkét játékos mindig nyerni akar) Us ellen, és megkérdezte tőled, hogy ezt kiderítsd.

### Input

Az első sor tartalmaz egy  $t$  egész számot ( $1 \leq t \leq 10$ ) — a résztesztek számát.

Minden teszt esetben:

Az első sor tartalmaz egy  $n$  egész számot ( $1 \leq n \leq 10^6$ ).

A második sor tartalmaz egy  $s$  karakterláncot, amelynek hossza  $n$ , és csak 0-9 közötti számjegyekből áll.

Garantált, hogy az összes részteszt  $n$ -jeinek összege nem haladja meg a  $10^6$ -ot.

### Output

Minden  $t$  sorra írd ki “Igen” ha Vus a kozák nyerhet; vagy “Nem” egyébként.

### Scoring

- (8 pont): a különböző számjegyek száma  $\leq 2$ ;
- (8 pont): a különböző számjegyek száma  $\leq 5$ ;
- (6 pont): a különböző számjegyek száma  $\leq 7$ ;
- (8 pont): csak egy számjegy jelenik meg többször;
- (7 pont): ha  $s_l = s_r, s_i = s_j$  és  $s_i \neq s_l$  ( $l \neq r; i \neq j$ ), akkor az  $[l, r]$  és  $[i, j]$  intervallumok nem fedik egymást;
- (7 pont):  $n \leq 4$ ;
- (6 pont):  $n \leq 8$ ;
- (6 pont):  $n \leq 12$ ;
- (6 pont):  $n \leq 15$ ;
- (38 pont): nincs további korlátozás.

A korlátozások minden  $t$  résztesztre vonatkoznak.

## Example

standard input	standard output
4	Yes
6	Yes
015423	No
7	No
1235212	
4	
1111	
6	
156156	

## Note

Az első példában két azonos számjegy soha nem lesz egymás mellett, mivel minden számjegy legfeljebb egyszer jelenik meg.

A második példában Vus eltávolíthatja az utolsó 2-t. Ezután, ha Us eltávolít egy 1-et vagy 2-t, Vus eltávolítja a 2-t vagy 1-et, így minden számjegy különbözővé válik, így Vus nyer. Ha azonban Us eltávolít egy 3-t vagy 5-t, akkor Vus először bármelyik 2-t eltávolítja, majd bármelyik 1-et.

A harmadik példában Us még a játék kezdete előtt nyer.

## Problem E. OldPost — NewPost

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

Trilyandia országában  $n$  város található, amelyeket  $n - 1$  út köt össze. Az ország úthálózata különleges:

- egy út csak két különböző várost köt össze;
- legfeljebb egy út lehet bármely két város között;
- bármely városból elérhető bármely más város, esetleg más városokon keresztül.

Az OldPost posta cég úgy döntött, hogy szállítási szolgáltatásokat nyújt ebben az országban. A korlátozott számú személyzet miatt a szállítás csak egy  $a$  útvonalon történik, amelynek hossza  $l$ , azaz a városok  $a_1, a_2, \dots, a_l$  sorozatán, amelyre az alábbiak érvényesek:

- $a_i \neq a_j$  (ha  $i \neq j$ );
- a  $a_i$  és  $a_{i+1}$  városokat egy út köti össze ( $1 \leq i < l$ ).

A profit maximalizálása érdekében az OldPost úgy döntött, hogy a lehető leghosszabb útvonalon működik.

Az OldPost ősi versenytársa — NewPost úgy döntött, hogy ő is ezt teszi, de a saját  $b$  útvonalán:  $b_1, b_2, \dots, b_l$ , amely nem feltétlenül különbözik az  $a$ -tól, de szintén a lehető leghosszabb.

A köztársasági elnök, miután értesült a két cég szándékairól, arra kérte az OldPostot és a NewPostot, hogy válasszanak olyan  $a$  és  $b$  útvonalakat, amelyek a lehető legtöbb várost lefedik (azaz biztosítják, hogy a lehető legtöbb város legalább egy útvonalon szerepel). Mivel Trilyandia elnöke a menedzsmentre specializálódott, míg a posta cégek az áruk szállítására, nem pedig az útvonalak keresésére, Önnek az a feladata, hogy megtalálja az ilyen  $a$  és  $b$  útvonalakat.

Vegye figyelembe, hogy mindkét cég a lehető leghosszabb útvonalat szeretné; még ha egy rövidebb útvonal választása növeli is a lefedett városok összegét.

Segítsen megtalálni az  $a$  és  $b$  útvonalakat, amelyek a lehető leghosszabbak és a lehető legtöbb várost fedik le.

### Input

Az első sor egyetlen egész számot tartalmaz  $n$  ( $2 \leq n \leq 6 \cdot 10^5$ ) — az ország városainak számát.

A következő  $n - 1$  sor mindegyike két egész számot tartalmaz  $x_i$  és  $y_i$  ( $1 \leq x_i, y_i \leq n, x_i \neq y_i$ ) — a városok párjait, amelyek között út található.

### Output

Az első sor egyetlen egész számot kell tartalmazzon  $l$  ( $2 \leq l \leq n$ ) — a megtalált kereskedelmi útvonalak hossza.

A második sor  $l$  egész számot kell tartalmazzon  $a_1, a_2, \dots, a_l$  ( $1 \leq a_i \leq n$ ) — az  $a$  útvonal városait.

A harmadik sor  $l$  egész számot kell tartalmazzon  $b_1, b_2, \dots, b_l$  ( $1 \leq b_i \leq n$ ) — a  $b$  útvonal városait.

Ha több optimális útvonal létezik, bármelyiket ki lehet adni.

### Scoring

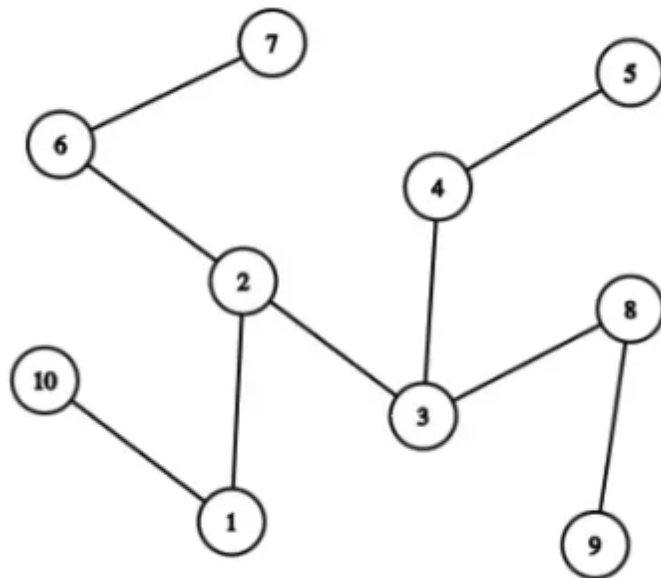
1. (8 pont):  $n \leq 10$ ;

- (8 pont):  $n \leq 20$ ;
- (6 pont):  $n \leq 1\,000$ ; pontosan egy város van, amely közvetlenül kapcsolódik az összes többihez;
- (8 pont):  $n \leq 20\,000$ ; pontosan két város van, amelyek csak egy másik várossal kapcsolódnak (a városok egy vonalat alkotnak);
- (8 pont):  $n \leq 20\,000$ ; pontosan egy város van, amely három vagy több várossal kapcsolódik;
- (8 pont):  $n \leq 10\,000$ ; legfeljebb 10 város van, amelyek csak egy másik várossal kapcsolódnak;
- (16 pont):  $n \leq 1\,000$ ;
- (18 pont):  $n \leq 70\,000$ ;
- (20 pont): nincs további korlátozás.

### Example

standard input	standard output
10	6
1 2	5 4 3 2 1 10
2 3	9 8 3 2 6 7
3 4	
4 5	
2 6	
6 7	
3 8	
8 9	
10 1	

### Note



A példában a leghosszabb lehetséges útvonalak a következők:



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

Az első és a negyedik útvonal lehetővé teszi az összes város lefedését.