# Átvágás

Van egy N csúcsú fánk. Kíváncsiak vagyunk, hogy minimálisan hány átvágás művelettel tudjuk bambusszá illetve csillaggá alakítani.

* Egy átvágás művelet során a fából eltávolítunk egy tetszőleges élet és egyik végpontját egy másik ponthoz kötjük úgy, hogy nem keletkezhet kör.
* Egy n csúcsú bambusz egy olyan fa, aminek n-2 darab 2 fokszámú és 2 darab 1 fokszámú csúcsa van.
* Egy n csúcsú csillag egy olyan fa, aminek n-1 darab 1 fokszámú és 1 darab n-1 fokszámú csúcsa van.

Írj programot, amely kiszámítja a minimálisan szükséges műveletek számát mind a bambusszá, mind a csillaggá alakításhoz!

## Bemenet

A *standard bemenet* első sorában a fa csúcsainak száma van (2≤N≤200000). A következő N-1 sor mindegyikében egy-egy él két végpontja van (1≤Ai≠Bi≤N).

## Kimenet

A *standard kimenet* első sorába az átalakításhoz minimálisan szükséges átvágás műveletek számát kell írni, elsőként ahhoz, hogy bambusszá, másodikként ahhoz, hogy csillaggá alakítsuk a bemenetben megadott fát!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 6 1 2 1 3 1 6 2 4 2 5 | 2 2 |
| Bemenet | Kimenet |
| 5 1 2 1 3 2 4 2 5 | 1 1 |

## Megjegyzés

Itt látható például 3 bambusz és 3 csillag:



Itt látható, egy lehetséges módszer, a példa fa bambusszá alakítására (az aktuális műveletben a pirossal áthúzott él kerül eltávolításra és a piros él kerül behúzásra):



## Korlátok

Időlimit: 0.7 s

Memórialimit: 64 MiB

# Kert

Egy kertben különböző színű virágokat ültettek. Azt mondjuk, hogy egy adott színű virág többségben van egy soron, ha a soron belül (nem feltétlenül egymás mellett) több van belőle, mint az összes többiből együttesen.

Írj programot, amely kiszámítja, hogy hány olyan sor van a kertben, amelyben valamely virág többségben van, valamint maximum hány azonos színű virág van soron belül egymás mellett!

## Bemenet

A *standard bemenet* első sorában a sorok száma (1≤M≤100), a sorokon belüli virágok száma (1≤N≤100000, M\*N≤500000) és a virágok színkódjainak lehetséges száma szerepel (1≤P≤100000000). A következő M sor mindegyikében N darab elültetett virág színkódja van (1≤Si≤P). Minden sorra érvényes, hogy a soron belül a legnagyobb és a legkisebb értékek közötti különbség legfeljebb 250000.

## Kimenet

A *standard kimenet* első sorába azon sorok számát kell írni, amelyben valamely virág többségben van! A második sorba az egymás melletti azonos színű virágok maximális száma kerüljön!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 4 7 9 2 1 2 3 8 2 2 4 7 2 4 9 7 4 5 5 2 **5 5 5** 7 2 3 2 3 2 3 1 | 2 3  Magyarázat: Az első sorban négy 2-es, a harmadikban öt 5-ös van. A leghosszabb azonos színű virágból álló rész a harmadik sorban a három 5-ös. |

## Korlátok

Időlimit: 0.5 mp.

Memórialimit: 32 MB

## Pontozás

A pontok 55%-a szerezhető olyan tesztekre, ahol P≤100000.

# Váltakozó szöveg

Váltakozónak nevezünk egy, az angol ábécé kisbetűiből álló szöveget hogyha semelyik két szomszédos karaktere nem egyezik meg. Például az "abcdeabcde" szöveg váltakozó, míg az "xyzzabc" nem.

Írj programot, amely megadja egy szövegkaraktereinek összekeverésével előállítható lexikografikusan minimális váltakozó szöveget!

## Bemenet

A *standard bemenet* első sorában a szöveg található (a hossza legfeljebb 200000).

## Kimenet

A *standard kimenet* első sorába egy a bemenetből előállítható váltakozó szöveg álljon, vagy a -1, ha nem lehet előállítani ilyet!!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| acaadddbbbb | abababdbdcd |

Megjegyzés

Két ugyanolyan hosszú, különböző szöveg közül az a lexikografikusan kisebb, amelyikben a közös kezdőszeletük után ábécésorrendben hamarabbi karakter következik.

* "abc" < "abd", mivel a közös kezdőszeletük az "ab" és az utána következő "c" hamarabb van az ábécében, mint a "d".
* "neeazz" < "neemes", mivel "a" hamarabb van az ábécében, mint "m".
* "ezzz" < "zeee", mivel "e" hamarabb van az ábécében, mint "z".

## Korlátok

Időlimit: 0.5 mp.

Memórialimit: 64 MiB

## Pontozás

A pontok 67%-a jár akkor, ha a kimeneti szöveg váltakozó.

A pontok maradék 33%-a jár akkor, ha a kimeneti szöveg lexikografikusan minimális is.

A pontok 16%-a jár olyan tesztekre, amikre a szöveg hossza legfeljebb 10.

A pontok további 16%-a jár olyan tesztekre, amikre a szövegben legfeljebb 2 különböző karakter van.

A pontok további 16%-a jár olyan tesztekre, amikre a szöveg hossza legfeljebb 2000.

# Vetélkedő

Egy internetes vetélkedőre csapatok jelentkezhetnek. A csapatokban páros számú résztvevő kell legyen (nem feltétlenül ugyanannyi). Minden csapat kap egy egyedi csapatazonosítót, ami 20 különböző karaktert tartalmazhat: számjegyeket és az angol abc kisbetűit a-tól j-ig (10 számjegy és 10 betű), legalább 3 karakteres. Egyik karakter sem szerepelhet egynél többször ugyanazon a csapatazonosítón belül és a csapatazonosítón belül a karakterek az ASCII kódok alapján növekvő sorrendben vannak (először a számjegyek szerepelnek, ha vannak, majd a kisbetűk, ha vannak, növekvő sorrendben). Csapaton belül a résztvevők azonosítói a csapat azonosítója karakterei tetszőlegesen összekeverve! A vetélkedőre páros számú résztvevő jelentkezett, de az egyik csapat egyik tagja sajnos nem tud részt venni a vetélkedőn, mert megbetegedett.

Írj programot, amely megadja azt a csapatazonosítót, amelyikhez tartozó csapatban az egyik tag nem tud részt venni a vetélkedőn!

## Bemenet

A *standard bemenet* első sorában a résztvevők száma van (1≤N≤99999, N páratlan). A következő N sor mindegyike egy versenyző legfeljebb 20 karakteres azonosítóját tartalmazza

## Kimenet

A *standard kimenet* első sorába azon csapatazonosítót kell írni, amelyből hiányzik egy játékos!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 5 9d6 8ac2 ica c28a d69 | aci  Magyarázat: Teljes (páros tagú) csapatot alkotnak a (9d6, d69) és (8ac2, c28a) azonosítók. Az ica azonosítójú résztvevő csapatából hiányzik egy ember. Ennek a csapatnak az azonosítója az aci. |

## Korlátok

Időlimit: 0.5 mp.

Memórialimit: 20 MB

## Pontozás

A pontok 40%-a szerezhető olyan tesztekre, amelyekben N<50000.