Mrav Mika je vredan mrav i mnogo voli da kopa rupe i pravi tunele. Toliko voli to da radi, da je iskopao čak *n* (5 ≤ *n* ≤ 1.000) rupa, koje su međusobno povezane sa *m* (1 ≤ *m* ≤ 10.000) tunela. Međutim, mrav Mika nije znao zlatno pravilo kopača rupa i tunela. Ono nalaže da je mogućnost podele rupa u dve grupe, tako da između grupa postoji bar ⌊ *m* / 2 ⌋ tunela, neophodan uslov za stabilnost podzemnog sistema a samim tim i dobijanje dozvole za korišćenje tunela. Inspekcija je došla kod Mike da proveri da li se pridržavao ovog pravila, a kako Mika ipak želi da svoje tunele pusti u rad, zanima ga da li je ono možda slučajno zadovo eno. Kako rupa i tunela ima previše, ovo je pretežak zadatak za Mikin mali mozak te vas je zamolio da mu pomognete. Dakle, potrebno je da nađete traženu podelu rupa u dve grupe (svaka rupa mora pripadati tačno jednoj grupi) ili da mu saopštite da neće moći da koristi tunele.

**Ulaz.**

(Ulazni podaci se učitavaju iz datoteke **mika.in**.) U prvom redu nalaze se brojevi *n* i *m*. Rupe su označene redom brojevima od 1 do *n*. U narednih m redova nalaze se po dva broja *a* i *b*, i oni označavauj da između rupa označenih brojevima *a* i *b* postoji direktan tunel. Između dve rupe može da postoji najviše jedan direktan tunel.

**Izlaz.**

(Izlazne podatke upisati u datoteku **mika.out**) Ukoliko nije moguće napraviti takvu podelu rupa, ispisati -1. U suprotnom, u prvom redu ispisati koliko ima rupa u prvoj i koliko u drugoj grupi, i potom u naredna dva reda redne brojeve rupa koje pripadaju prvoj, odnosno drugoj grupi.

**Primer 1.**

|  |  |  |
| --- | --- | --- |
| **mika.in** |  | **mika.out** |
| 8 10  1 2  1 5  1 6  5 6  2 6  3 8  3 7  3 4  4 8  7 8 |  | 4 4  5 6 7 8  1 2 3 4 |

**Primer 2.**

|  |  |  |
| --- | --- | --- |
| **mika.in** |  | **mika.out** |
| 5 5  1 2  2 3  3 4  4 5  5 1 |  | 3 2  3 4 5  1 2 |

Robotić WALL-E se još od Okružnog takmičenja usamljen igra na deponiji smeća i to mu već pomalo postaje dosadno. Kako bi stao na put dokolici, priredio je veliku žurku na deponiji na koju je pozvao sve svoje metalne drugare sa fejsbuka. Robotska žurka se odvija standardno: Svaki robotić zauzme jedno polje matrice koja predstavlja deponiju i zamisli jedan od četiri moguća pravca u kom želi da načini korak. Potom se robotići istovremeno pomere, svaki za tačno jedan korak u svom željenom pravcu. Tako se nađu na novim pozicijama i žurka se završava (sve što je dobro traje kratko).

Međutim, odaziv je bio veoma velik i na žurci se napravila prilična gužva. Zbog toga nisu svi robotići u stanju da načine željeni korak. Da bi robotić uspeo da ode na polje na koje je zamislio, ni jedan drugi robotić se ne sme naći na tom polju nakon odigranog koraka. To znači da od svih robotića koji žele da dođu na neko polje samo jedan može da uspe u tome, i to samo ako robotić koji je prethodno bio na tom polju uspe da ga napusti. U toku koraka robotići se mogu mimoilaziti bez problema, ali po izvršenju koraka na svakom polju se sme nalaziti najviše jedan robotić.

WALL-E želi da mu žurka koliko-toliko uspe i da usreći što više svojih drugara. On treba da odabere najveći mogući broj robotića koji će moći da naprave korak, dok će svi ostali morati da ostanu na svojim pozicijama. Koliko najviše robotića može napraviti korak?

**Ulaz.**

(Ulazni podaci se učitavaju iz datoteke **robotici.in**.) U prvom redu nalaze se dva broja *m* i *n* (1 ≤ *m*, *n* ≤ 200), dimenzije matrice koja predstavlja deponiju. U svakom od sledećih *m* redova nalazi se *n* razmakom razdvojenih brojeva koji predstavljaju polja. Brojevi označavaju šta se nalazi na odgovarajućem polju i imaju sledeća značenja:

1. 0 - prazno polje
2. 1 - robotić koji želi da napravi korak na desno
3. 2 - robotić koji želi da napravi korak na gore
4. 3 - robotić koji želi da napravi korak na levo
5. 4 - robotić koji želi da napravi korak na dole

Nijedan robotić neće poželeti da napusti deponiju za vreme žurke (tj. napravi korak van matrice).

**Izlaz.**

Na izlaz ispisati samo jedan broj - koliko najviše robotića moće da načini korak.

**Primer 1.**

|  |  |  |
| --- | --- | --- |
| **robotici.in** |  | **robotici.out** |
| 4 5  0 0 0 0 0  0 1 1 4 0  0 2 0 1 0  0 0 0 0 0 |  | 5 |

**Primer 2.**

|  |  |  |
| --- | --- | --- |
| **robotici.in** |  | **robotici.out** |
| 4 3  0 0 0  0 4 0  0 2 0  0 0 0 |  | 2 |

Cica je vođa grupe progsica - čirlidersica koje su se specijalizovale za zabavljanje publike na Državnom takmičenju iz programiranja. Tokom poluvremena ove prestižne manifestacije progsice izvode svoj performans.

Svaka progsica obučena je u kostim određene boje. Cica i ekipa zamislile su da se, kao vrhunac nastupa, poređaju u liniju duž pravca sever-jug, a da potom neke od njih napuste arenu (moguće je i da nijedna ne ode) i to tako da publika sa istočne tribine, gledajući preostale progsice, vidi isti redosled boja kakav vidi i publika sa zapadne tribine. Tim povodom vas mole za pomoć: pitaju na koliko načina mogu ostvariti svoju zamisao.

**Ulaz.**

(Ulazni podaci se učitavaju iz datoteke **progsice.in**.) U prvom redu ulazne datoteke nalazi se broj progsica, *n* (1 ≤ *n* ≤ 5.000). U drugom redu dat je poredak progsica, onako kako ih vidi publika s istočne tribine, u momentu kada neke od njih treba da napuste arenu. Poredak je dat u formi stringa dužine *n*, pri čemu *i*-ti karakter u stringu označava boju kostima *i*-te progsice (naravno, isti karakteri uvek označavaju istu boju, a različiti karakteri različitu).

**Izlaz.**

(Izlazne podatke upisati u datoteku **progsice.out**) U prvi i jedini red izlazne datoteke upisati ostatak pri deljenju broja načina da progsice ostvare svoju zamisao brojem 1.000.000.007.

**Primer 1.**

|  |  |  |
| --- | --- | --- |
| **progsice.in** |  | **progsice.out** |
| 5  patka |  | 8 |

**Objašnjenje.**

Na slici su prikazane sve mogućnosti (prekrižene su one progsice koje treba da napuste arenu). Slika prikazuje pogled s istočne tribine, a uočava se da i publika sa zapadne vidi isti redosled boja. Primetiti da se ne ubraja mogućnost kada sve progsice napuste teren (tj., mora ostati bar jedna da nastavi zabavljanje publike).