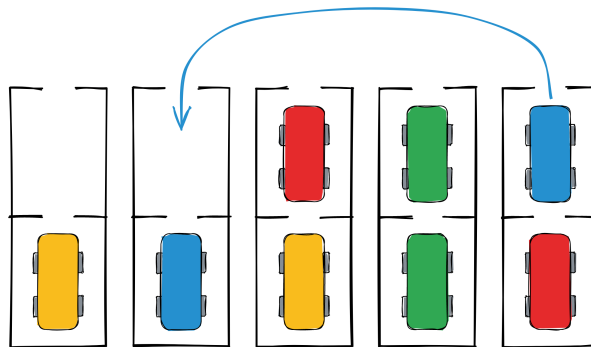


Zadatak: Parking

Nikola radi kao parkirač u otmjenom restoranu. Njegov posao je čekati dolazak uvažanih gostiju poput Naomi, uljudno ih pozdraviti, preuzeti ključeve njihovih vozila te ih parkirati na obližnje parkiralište. Jednom kada događaj završi, on se treba pobrinuti da svaki gost dobije natrag njegovo vozilo te da sretan napusti mjesto.

Jednu večer, nedugo nakon što je završio s parkiranjem svih vozila, primijetio je dosta zanimljivo svojstvo u vezi njihovih boja. Ispostavilo se da je bilo točno $2N$ vozila na parkiralištu i da su bila obojena u N različitih boja na način da su bila točno po dva vozila svake boje. Boje vozila označavamo prirodnim brojevima od 1 do N .

Samo parkiralište možemo prikazati kao niz od M parkirnih mjesta, označenih prirodnim brojevima od 1 do M , gdje svako parkirno mjesto može sadržavati najviše dva vozila. Postoji samo jedan ulaz na parkirno mjesto, a vozilo može ući ili izaći iz mjesta ako ni jedno drugo vozilo ne krči ulaz. Nazovimo vozilo parkirano bliže ulazu *gornjim vozilom*, a vozilo parkirano dalje ulazu *donjim vozilom*. Nikola je parkirao vozila na način da je svako parkirno mjesto prazno, puno ili sadrži jedno donje vozilo.



Prikaz prvog probnog primjera koje pokazuje jedinu moguću prvu vožnju.

Nikola želi reparkirati vozila tako da je svaki par vozila iste boje parkiran na isto parkirno mjesto. Baš ga briga koje će mjesto sadržavati koju boju i koje će vozilo biti gornje, a koje donje (Nikola nema gore/dolje preference). On će reparkirati vozila kroz niz *vožnji*. U svakoj vožnji, on će sjesti u parkirano vozilo koje ima mogućnost izlaska iz trenutnog parkirnog mjesta te će ga odvesti na drugo parkirno mjesto koje je ili prazno (u tom slučaju ga parkira kao donje vozilo) ili sadrži jedno parkirano vozilo **iste boje** kao ono koje trenutno vozi (u tom slučaju ga parkira kao gornje vozilo).

Nikola želi minimizirati broj vožnji potreban da reparkira sva vozila prema njegovim željama. Tvoj je zadatak pomoći mu tako da nađeš najkraći niz vožnji koji bi postigao traženi cilj ili odrediti da takav niz ne postoji.

Učit

The first line contains two space-separated integers N and M from the task description.

The i -th of the next M lines contains two space-separated integers b_i and t_i ($0 \leq b_i, t_i \leq N$) that describe the i -th parking space. More precisely, the number b_i represents the bottom vehicle color, and the number t_i represents the top vehicle color. If a position in the parking space is empty, the corresponding integer will be equal to 0. It is guaranteed that no parking space contains only a top vehicle, i.e. if $b_i = 0$, then $t_i = 0$ as well.



Iščit

If there is no sequence of drives that could repark the vehicles according to Nikola's wishes, print -1 in the only line.

Otherwise, the first line should contain an integer K , the smallest number of drives needed to satisfy Nikola's goal.

The i -th of the next K lines should describe the i -th drive. More precisely, it should contain two integers, x_i and y_i ($1 \leq x_i, y_i \leq M, x_i \neq y_i$), representing that Nikola should take a vehicle from parking space x_i to parking space y_i in the i -th drive. Of course, the x_i -th parking space must contain at least one vehicle at that point, and the vehicle closer to the entrance must be movable to parking space y_i , i.e. parking space y_i must be either empty or contain a vehicle of the same color.

Bodovanje

In all subtasks it holds that $1 \leq N \leq M \leq 200\,000$.

If your solution correctly determines the smallest number of drives in all test cases of a certain subtask, but incorrectly outputs the description of drives in some of them (or doesn't output it at all), it will receive 20% of the points allocated to that particular subtask.

Subtask	Score	Constraints
1	10	$M \leq 4$
2	10	$2N \leq M$
3	25	All parking spaces are initially either empty or full, and $N \leq 1\,000$.
4	15	All parking spaces are initially either empty or full.
5	25	$N \leq 1\,000$
6	15	No additional constraints.

Probni primjeri

učit

4 5
1 0
2 0
1 3
4 4
3 2

iščit

3
5 2
3 5
3 1

učit

4 5
0 0
2 1
3 1
3 4
2 4

iščit

-1

učit

5 7
1 0
2 1
2 3
4 3
5 4
5 0
0 0

iščit

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



Clarification of the first example: The image from the task description depicts the initial state of the parking lot in this example. Notice that in this case, each drive is forced, i.e. there is only one valid first drive, only one valid second drive, and two equivalent third drives, after which we reach the end goal.