



## Croatian Olympiad in Informatics

October 3<sup>rd</sup> 2020

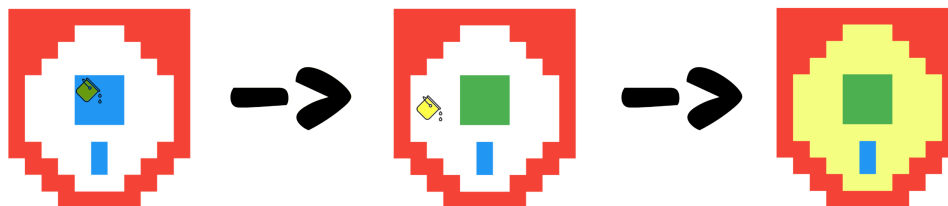
### Tasks

Task	Time Limit	Memory Limit	Points
<b>Paint</b>	3 seconds	512 MiB	100
<b>Pastiri</b>	1 sekunda	512 MiB	100
<b>Semafor</b>	4 sekunde	512 MiB	100
<b>Zagrade</b>	3 sekunde	512 MiB	100
<b>Total</b>			400



## Task Paint

We will represent the drawing area of *MS Paint* as a rectangular grid of unit squares divided into  $R$  rows and  $S$  columns. Each square of the grid represents a single pixel that can be colored in one of the  $10^5$  different colors. When the user applies the so called *bucket tool* with color  $A$  on a pixel  $(r, s)$  which is colored in the color  $B$ , then all pixels in the *monochrome neighborhood* of pixel  $(r, s)$  change their color to  $A$ . Monochrome neighborhood of a pixel  $(r, s)$  is a set of pixels that are reachable by *walking* from  $(r, s)$  in the four general directions (up, down, left and right) without changing the color of the pixel along the way. Note that the pixel  $(r, s)$  is itself a part of its monochrome neighborhood.



You are given a starting image drawn in *MS Paint* along with  $Q$  instructions that should be executed in the given order. Each instruction tells you on which pixel should you apply the bucket tool and with what color. Your task is to how the image looks like after all instructions are executed.

### Input

The first line contains integers  $R$  and  $S$  from the task description.

Each of the next  $R$  lines contains  $S$  non-negative integers less than 100 000 that represent the starting image drawn in *MS Paint*. More precisely, the  $j$ -th number in the  $i$ -th row of the image represents the color of the pixel  $(i, j)$ .

The next line contains an integer  $Q$  from the task description.

The  $i$ -th of the next  $Q$  lines contains integers  $r_i$ ,  $s_i$  and  $c_i$  ( $1 \leq r_i \leq R, 1 \leq s_i \leq S, 0 \leq c_i < 100\,000$ ), which represent the  $i$ -th instruction that tells you to use the bucket tool with color  $c_i$  on the pixel  $(r_i, s_i)$ .

### Output

You should output the final state of the image in the same format as it was given in the input.

### Scoring

Subtask	Score	Constraint
1	8	$1 \leq R \cdot S \leq 10\,000, 1 \leq Q \leq 10\,000$
2	9	$R = 1, 1 \leq S \leq 200\,000, 1 \leq Q \leq 100\,000$
3	31	$1 \leq R \cdot S \leq 200\,000, 1 \leq Q \leq 100\,000$ Each pixel will in every moment be colored either in color 0 or color 1.
4	52	$1 \leq R \cdot S \leq 200\,000, 1 \leq Q \leq 100\,000$



## Examples

input

```
12 11
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 1 1 1 1
1 1 1 0 0 0 0 0 1 1 1
1 1 0 0 0 0 0 0 0 1 1
1 0 0 0 2 2 2 0 0 0 1
1 0 0 0 2 2 2 0 0 0 1
1 0 0 0 2 2 2 0 0 0 1
1 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 2 0 0 0 1 1
0 1 1 0 0 2 0 0 1 1 0
0 0 1 1 0 0 0 1 1 0 0
0 0 0 1 1 1 1 1 0 0 0
2
5 5 3
6 2 4
```

output

```
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 4 4 4 1 1 1 1
1 1 1 4 4 4 4 4 1 1 1
1 1 4 4 4 4 4 4 4 1 1
1 4 4 4 3 3 3 4 4 4 1
1 4 4 4 3 3 3 4 4 4 1
1 4 4 4 3 3 3 4 4 4 1
1 4 4 4 4 4 4 4 4 4 1
1 1 4 4 4 2 4 4 4 1 1
0 1 1 4 4 2 4 4 1 1 0
0 0 1 1 4 4 4 1 1 0 0
0 0 0 1 1 1 1 1 0 0 0
```

input

```
4 4
1 0 1 3
1 3 2 2
3 3 1 2
2 2 1 3
3
1 2 3
3 2 1
4 2 3
```

output

```
1 1 1 3
1 1 2 2
1 1 1 2
3 3 1 3
```

input

```
6 6
1 2 1 2 2 2
3 1 2 1 3 1
3 3 2 3 2 2
2 3 1 3 3 2
3 3 3 3 3 3
2 3 2 2 2 1
4
```

```
6 2 2
3 5 2
3 2 3
1 2 3
```

output

```
1 3 1 2 2 2
3 1 3 1 3 1
3 3 3 3 3 3
3 3 1 3 3 3
3 3 3 3 3 3
3 3 3 3 3 1
```

**Clarification of the first example:** The figure from the task description corresponds to the input of the first example. White color corresponds to number 0, red color corresponds to number 1, blue color corresponds to number 2, green color corresponds to number 3 and yellow color corresponds to number 4.



## Task Pastiri

„Nikada nisam bio toliko sit da nisam mogao pojesti još kilu janjetine.” – gospodin M.

Stado od  $K$  ovaca živi na stablu, tj. jednostavnom povezanom grafu bez ciklusa koji se sastoji od  $N$  vrhova označenih prirodnim brojevima od 1 do  $N$ . U svakom vrhu stabla živi najviše jedna ovca. Stara narodna mudrost kaže: „vukovi se kriju i ~~piju~~ jedu ti ovce.”

Kako bismo obranili naše ovce od zlih vukova, potrebno je u neke vrhove stabla postaviti pastire, tako da svaku ovcu čuva barem jedan pastir. Poznato je da **svaki pastir čuva sve sebi najbliže ovce**, i samo njih. Udaljenost između neke ovce i nekog pastira jednaka je broju vrhova na jedinstvenom putu između vrha na kojem se nalazi ta ovca i vrha na kojem se nalazi taj pastir (uključivo). Također, pastir se može nalaziti u istom vrhu kao i neka ovca, dakako, tada on čuva samo tu ovcu.

Odredite **najmanji broj pastira** koje je potrebno postaviti u vrhove danog stabla tako da **svaku ovcu čuva barem jedan pastir**. Dodatno, odredite neki takav raspored najmanjeg mogućeg broja pastira.

### Ulazni podaci

U prvom su retku prirodni brojevi  $N$  i  $K$  ( $1 \leq K \leq N$ ) iz teksta zadatka.

U sljedećih  $N - 1$  redaka nalaze se po dva prirodna broja  $a_i$  i  $b_i$  ( $1 \leq a_i, b_i \leq N$ ) koji označavaju da postoji neusmjerena veza u stablu između vrhova s oznakama  $a_i$  i  $b_i$ .

U sljedećem se retku nalazi  $K$  različitih prirodnih brojeva  $o_i$  ( $1 \leq o_i \leq N$ ) koji predstavljaju oznake čvorova u kojima se nalaze ovce.

### Izlazni podaci

U prvom retku ispišite broj  $X$ , koji predstavlja najmanji mogući broj pastira iz teksta zadatka.

U drugom retku ispišite  $X$  brojeva odvojenih razmakom, koji predstavljaju oznake čvorova u koje treba postaviti pastire.

### Bodovanje

Podzadatak	Broj bodova	Ograničenja
1	8	$1 \leq N \leq 500\,000$ , svaki vrh $x = 1, \dots, n - 1$ je povezan s vrhom $x + 1$
2	18	$1 \leq K \leq 15$ , $1 \leq N \leq 500\,000$
3	23	$1 \leq N \leq 2\,000$
4	51	$1 \leq N \leq 500\,000$



## Probni primjeri

input

```
4 2
1 2
2 3
3 4
1 4
```

output

```
2
1 3
```

input

```
9 5
1 2
2 3
3 4
3 5
1 6
1 7
7 8
8 9
2 5 6 7 9
```

output

```
3
1 4 9
```

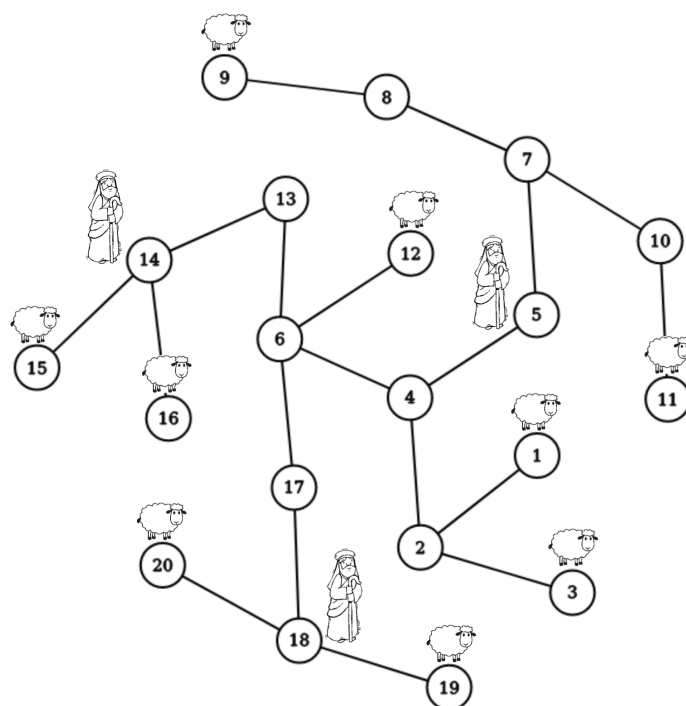
input

```
20 9
1 2
2 3
2 4
4 5
4 6
5 7
7 8
8 9
7 10
10 11
6 12
6 13
6 17
13 14
14 15
14 16
17 18
18 19
18 20
1 3 9 11 12 15 16 19 20
```

output

```
3
5 14 18
```

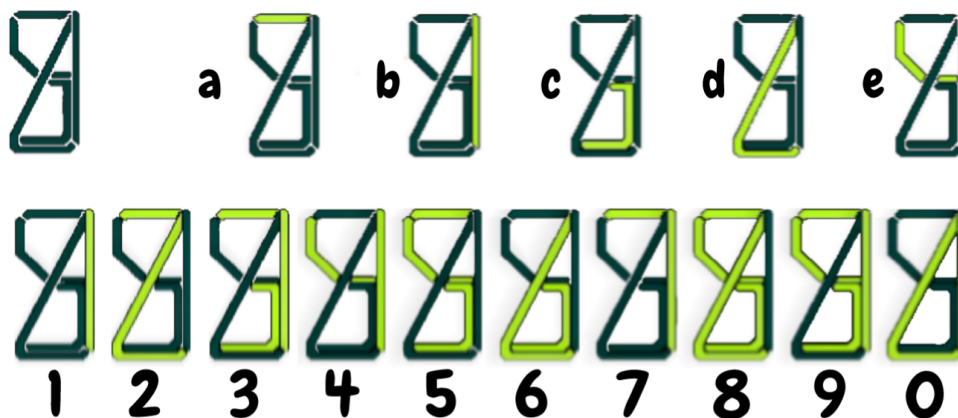
Pojašnjenje trećeg probnog primjera:





## Task Semafor

Vjerojatno ste se više puta u životu susreli sa tzv. *7-segmentnim* displejom koji se koristi za prikaz znamenaka na raznim digitalnim uređajima poput satova ili kalkulatora. Zbog svoje jednostavnosti, intuitivnosti i estetske ugođe, takav je dizajn prihvaćen diljem planete. Ipak, mladi Matej smatra ga *polu-proizvodom* te tvrdi da se ista funkcionalnost može ostvariti na mnogo efikasniji način, koristeći svega pet segmenata.



Dizajn 5-segmentnog displeja – segmenti su označeni slovima od a do e.

Prvi poduzetnički korak odlučio je napraviti u najprosperitetnijoj grani hrvatske industrije – nogometu. Svoj revolucionarni dizajn iskoristit će pri izradi semafora za izmjene igrača tijekom utakmica 1. HNL, a trenutno radi na prezentaciji koju će iznijeti čelnicima Hrvatskog nogometnog saveza. Semafor se sastoji od  $M$  displeja koji (slijeva nadesno) predstavljaju znamenke broja koji nosi nogometaš koji treba izaći, odnosno ući u teren. Na početku Matejeve prezentacije na semaforu će se nalaziti broj  $X$ , a Matej će svake sekunde napraviti jedan od sljedećih poteza:

- Upalit će neki od ugašenih segmenata.
- Ugasit će neki od upaljenih segmenata.

Također, Matej će pritom osigurati da se nakon svakog  $K$ -tog poteza na semaforu nalazi ispravno napisan broj. Ukupno će Matej napraviti  $N$  poteza, a na kraju prezentacije (nakon  $N$ -tog poteza) semafor će također predstavljati ispravno napisan broj. Broj je ispravno napisan ako je svaka njegova znamenka ispravno napisana (kako je prikazano na slici). Također, brojevi koji imaju manje od  $M$  znamenaka trebaju sadržavati odgovarajući broj početnih nula.

Za svako završno stanje (cijeli broj između 0 i  $10^M - 1$ ), Mateja zanima na koliko je različitih načina mogao vući poteze da se na kraju prezentacije na semaforu nalazi upravo taj broj te da su pritom zadovoljeni svi uvjeti iz prethodnog odlomka, ako se na početku prezentacije na semaforu prikazivao broj  $X$ . Dva načina (niza poteza) smatramo različitim ako, kada bismo poteze istovremeno vukli na dva semafora, postoji trenutak u kojem semafori ne prikazuju identično stanje. Budući da broj načina može biti poprilično velik, potrebno je ispisati njegov ostatak pri dijeljenju s  $10^9 + 7$ .

### Ulazni podaci

U prvom su retku prirodni brojevi  $M$ ,  $N$ ,  $K$  ( $1 \leq K \leq N$ ) i  $X$  ( $0 \leq X < 10^M$ ) iz teksta zadatka.



## Izlazni podaci

U  $i$ -tom od  $10^M$  redaka izlaza treba se nalaziti broj različitih načina da semafor na kraju prezentacije prikazuje broj  $i - 1$ . Brojeve treba ispisati modulo  $10^9 + 7$ .

## Bodovanje

Podzadatak	Broj bodova	Ograničenja
1	6	$M = 1, 1 \leq N \leq 12$
2	15	$M = 1, 1 \leq N \leq 10^{15}$
3	12	$M = 2, 1 \leq N \leq 1\,500, K = N$
4	12	$M = 2, 1 \leq N \leq 10^{15}, 1 \leq K \leq 15$
5	15	$M = 2, 1 \leq N \leq 10^{15}, 1 \leq K \leq 1\,500$
6	40	$M = 2, 1 \leq N \leq 10^{15}$

## Probni primjeri

input

1 2 1 5

output

0  
0  
0  
1  
0  
2  
0  
0  
0  
0

input

1 3 3 8

output

0  
0  
0  
6  
0  
13  
0  
0  
0  
0

input

1 4 2 4

output

24  
0  
8  
0  
37  
0  
4  
28  
4  
24

**Pojašnjenje prvog probnog primjera:** Na početku prezentacije se na (jednoznamenkastom) semaforu nalazi broj 5 te se nakon svakog poteza (zbog  $K = 1$ ) na semaforu treba nalaziti ispravno napisan broj. Matej će za vrijeme prezentacije napraviti ukupno  $N = 2$  poteza. Na završetku prezentacije na semaforu se mogu nalaziti samo brojevi 3 ili 5. Broj 3 možemo dobiti na jedan način ( $5 - 9 - 3$ ) dok broj 5 možemo dobiti na dva načina ( $5 - 9 - 5$  i  $5 - 8 - 5$ ).



## Task Zagrade

Središnja obavještajna agencija (engl. *Central Intelligence Agency, CIA*) je obavještajna služba Sjedinjenih Američkih Država. U njoj se nadležnosti primarno nalazi prikupljanje i analiza podataka o stranim vladama, korporacijama i pojedincima. Nedvojbeno je da CIA prikuplja i analizira ogromne količine računalnih lozinki domaćih i stranih građana te razvija alate kojima je moguće kompromitirati još veći broj korporacija i pojedinaca.

Vaš je zadatak jednostavan, kompromitirajte sigurnost središnje obavještajne agencije. Sretno!

Naravno, budući da su im poznati tipični obrasci izrade računalnih lozinki, pokušaji poput `123456`, `password`, `1q2w3e4r` ili `welcome` neće uroditi plodom. Srećom, tajnim smo kanalima ipak otkrili neke informacije koje bi vam mogle pomoći.

Naime, njihova se glavna lozinka sastoji od  $N$  znakova, pri čemu je  $N$  paran te je točno polovina znakova jednaka otvorenoj zagradi (`'( '`), dok je druga polovina znakova jednaka zatvorenoj zagradi (`' ) '`). Također, umjesto uobičajene “*zaboravili ste lozinku?*” funkcionalnosti, inženjeri su odlučili zaboravljivom administratoru dopustiti da najviše  $Q$  puta postavi pitanje oblika: “*je li interval lozinke koji započinje  $a$ -tim, a završava  $b$ -tim znakom matematički validan?*”.

Matematičku validnost niza zagrada definiramo induktivno na sljedeći način:

- `()` je matematički validan niz zagrada.
- Ako je  $A$  matematički validan niz zagrada, tada je i  $(A)$  matematički validan niz zagrada.
- Ako su  $A$  i  $B$  matematički validni nizovi zagrada, tada je i  $AB$  matematički validan niz zagrada.

## Interakcija

Ovo je interaktivni zadatak. Vaš program treba uspostaviti dijalog sa programom izrađenim od strane organizatora koji simulira funkcionalnost **fiktivnog** nesigurnog poslužitelja središnje obavještajne agencije iz teksta zadatka.

Prije interakcije vaš program treba sa standardnog ulaza pročitati paran prirodan broj  $N$  koji predstavlja duljinu tajne lozinke te prirodan broj  $Q$  koji predstavlja broj upita koje vaš program smije poslati.

Nakon toga, vaš program može slati upite pisanjem na standardni izlaz. Svaki upit treba biti ispisan u zaseban redak te poprimati oblik “`? a b`” gdje vrijedi  $1 \leq a \leq b \leq N$ . Nakon svakog ispisanog upita, vaš program treba napraviti *flush* izlaza te sa standardnog ulaza pročitati *odgovor* na upit – broj 1 ako je interval lozinke koji započinje  $a$ -tim, a završava  $b$ -tim znakom matematički validan, odnosno 0 ako to nije. Vaš program smije poslati najviše  $Q$  ovakvih upita.

Kada je vaš program odgonetnuo tajnu lozinku, treba na standardni izlaz ispisati redak oblika “`! x1x2...xN`” gdje znakovi  $x_1, x_2, \dots, x_N$  predstavljaju znakove tajne lozinke. Nakon toga, vaš program ponovno treba napraviti *flush* izlaza i završiti izvođenje.

**Napomena:** Putem sustava za evaluaciju možete preuzeti primjere izvornih kodova koji na ispravan način obavljaju interakciju, uključujući *flush* izlaza.





## Bodovanje

Podzadatak	Broj bodova	Ograničenja
1	14	$1 \leq N \leq 1\,000$ , $Q = \frac{N^2}{4}$ , cijela lozinka je matematički validan niz.
2	7	$1 \leq N \leq 1\,000$ , $Q = \frac{N^2}{4}$
3	57	$1 \leq N \leq 100\,000$ , $Q = N - 1$ , cijela lozinka je matematički validan niz.
4	22	$1 \leq N \leq 100\,000$ , $Q = N - 1$

## Primjer interakcije

Izlaz	Ulaz	Napomena
	6 9	Tajna lozinka je ((())), duljine 6, a program smije postaviti najviše 9 upita.
? 1 6	1	Cijela lozinka je matematički validan niz zagrada.
? 1 2	0	(( nije matematički validan niz zagrada.
? 2 4	0	(( nije matematički validan niz zagrada.
? 2 5	1	(( )) je matematički validan niz zagrada.
? 3 4	1	() je matematički validan niz zagrada.
! ((()))		Lozinka je uspješno odgonetnuta i CIA je kompromitirana.