

Igra u učionici

Istaknuta srednja škola u Taškentu slavi svoju godišnjicu organizacijom utakmice između učenika i nastavnika. U učionici sjedi N učenika, numeriranih od 0 do $N - 1$. Svaki učenik drži komad papira koji sadrži niz cijelih brojeva. U početku je svaki papir prazan, tako da sadrži prazan niz.

Učenici igraju igru sa M nastavnika, numerisanim od 0 do $M - 1$, i direktorom škole. Igra se igra u M koracima, također numerisanim od 0 do $M - 1$. U koraku j , nastavnik j ulazi u učionicu i dešavaju se sljedeći događaji:

- Neki (moguće nijedan) učenik diže ruku. Garantuje se da u svakom koraku barem jedan učenik **ne** diže ruku, a svaki učenik može dići ruku **najviše jednom** tokom cijele igre.
- Nastavnik pregleda papire koje učenici trenutno drže i može **modifikovati** papire koje drže učenici koji **nisu** digli ruku u ovom koraku. Za svakog takvog učenika, nastavnik može zamijeniti sadržaj njihovog papira novim (moguće praznim) nizom cijelih brojeva (integers). Svaki cijeli broj mora biti između 0 i 63, uključujući i te brojeve, a niz može imati najviše 63 elemenata.
- Nakon ovih modifikacija, nastavnik j odlazi, a učenici razmjenjuju svoje papire prema tajnoj permutaciji P . To jest, svaki učenik i ($0 \leq i < N$) daje svoj papir učeniku $P[i]$, gdje su $P[0], P[1], \dots, P[N - 1]$ **različitih** cijelih brojeva između 0 i $N - 1$, uključujući i te brojeve. Vrijednosti P su **fiksne** tokom svih koraka igre, ali nastavnici i direktor ih ne znaju.

Nakon što su svi M koraci završeni i posljednja zamjena prema P izvršena, ulazi direktor. Gledajući samo papire koje drže učenici, direktor mora odrediti, za svakog učenika i ($0 \leq i < N$), broj koraka j u kojem je učenik i podigao ruku ili zaključiti da učenik i nikada nije podigao ruku.

Nastavnicima nije dozvoljena međusobna komunikacija niti komunikacija sa direktorom, osim putem nizova cijelih brojeva zapisanih na papirima. Svaki nastavnik zna u kojem trenutku (koraku) ulazi u učionicu

Tvoj zadatak je da osmisliš i implementiraš strategiju za nastavnike i direktora škole koja će ispravno identifikovati da li je i kada svaki učenik podigao ruku. Tvoj rezultat će zavisiti od maksimalne dužine bilo kojeg niza napisanog na papiru: što je kraća maksimalna dužina donosi isti ili veći broj bodova.

Detalji implementacije

Potrebno je implementirati dvije procedure, jednu za nastavnike i jednu za direktora škole.

Procedura koju biste trebali primijeniti za **nastavnike** je sljedeća:

```
std::vector<std::vector<int>> process_step(  
    int N, int M, int R,  
    std::vector<int> T,  
    std::vector<std::vector<int>> A)
```

- N : broj studenata.
- M : broj koraka, a takođe i broj nastavnika.
- R : trenutni broj koraka (od 0 do $M - 1$).
- T : (moguće prazan) niz koji se sastoji od indeksa studenata koji su digli ruku u ovom koraku.
- A : niz dužine N koji opisuje papire, gdje je $A[i]$ ($0 \leq i < N$) niz cijelih brojeva na papiru koji drži student i na početku koraka.
- Ova procedura se poziva tačno M puta po igri, redom $R = 0, 1, \dots, M - 1$.

Procedura treba da vrati niz B dužine N , koji predstavlja radove nakon nastavnikovih izmjena, u istom formatu kao A .

- Za svakog studenta i koji je podigao ruku (tj. i se pojavljuje u T), papir se ne smije mijenjati, tako da $B[i] = A[i]$.
- Za svaki i takav da je $0 \leq i < N$, dužina $B[i]$ ne smije biti veća od 63, a svaki cijeli broj u $B[i]$ mora biti između 0 i 63, uključujući i te brojeve.

Procedura koju biste trebali primijeniti za **direktora** je sljedeća:

```
std::vector<int> determine_steps(  
    int N, int M, std::vector<std::vector<int>> A)
```

- N, M : isto kao gore.
- A : niz dužine N , gdje je $A[i]$ niz cijelih brojeva na papiru koji drži student i nakon svih M koraka.
- Ova procedura se poziva tačno jednom po igri, nakon posljednjeg poziva funkcije `process_step`.

Procedura mora vratiti niz D dužine N . Za svaki i takav da je $0 \leq i < N$, mora važiti da

- $D[i] = j$ ako je učenik i podigao ruku tokom koraka j , ili
- $D[i] = -1$ ako učenik i nije digao ruku tokom cijele igre.

Vašem programu nije dozvoljeno čuvanje ili prenos bilo kakvih informacija iz jednog poziva funkcije `process_step` u drugi, osim putem povratne vrijednosti funkcije `process_step`. Ako vaš program pokuša ovo učiniti, vaš rezultat u CMS-u može biti netačan i može se smanjiti nakon završetka takmičenja. Imajte na umu da ocjenjivač može istovremeno izvršiti više instanci vašeg programa. To znači da se pozivi funkcija `process_step` i `determine_steps` iz iste igre

mogu izvršiti u različitim instancama, a pozivi funkcija `process_step` i `determine_steps` iz različitih igara mogu se izvršiti u istoj instanci. Svaki testni slučaj uključuje do 5 igara.

Ograničenja

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Svaki učenik podiže ruku **najviše jednom** tokom cijele igre. Drugim riječima, za svakog učenika i postoji najviše jedan korak j u kojem i podiže ruku.
- Na svakom koraku, barem jedan učenik **ne** diže ruku.

Podzadaci i bodovanje

Podzadatak	Rezultat	Dodatna ograničenja
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ za sve $0 \leq i \leq N - 1$.
4	25	Najviše jedan student diže ruku na svakom koraku.
5	56	Nema dodatnih ograničenja.

Za svaki testni slučaj, vaš rezultat je 0 (prijavljeno kao `Output isn't correct` u CMS-u) ako je povratna vrijednost bilo kojeg poziva procedure `process_step` nevažeća ili ako je povratna vrijednost bilo kojeg poziva procedure `determine_steps` netačna.

U suprotnom, neka C bude maksimalna dužina **bilo kojeg** niza u **bilo kojem** B koji vraća `process_step`. Tada se rezultat za testni slučaj u podzadatku s rezultatom S izračunava kao $S \cdot X$, gdje X zavisi od C prema sljedećoj tabeli:

Stanje	X
$C \leq 2$	1.00
$C = 3$	0.75
$C = 4$	0.55
$5 \leq C \leq 13$	$0.50 - 0.03 \cdot (C - 5)$
$14 \leq C \leq 63$	$0.19 \cdot \frac{64-C}{64-14} + 0.04$

Primjer

Razmotrimo scenario sa $N = 4$ učenika, $M = 2$ nastavnika i permutacijom $P = [0, 3, 1, 2]$. U početku su svi papiri prazni.

Ocjenjivač prvo poziva:

```
process_step(4, 2, 0, [1], [], [], [], [])
```

Učenik 1 je podigao ruku, tako da se njegov rad ne može mijenjati. Nastavnik 0 može odlučiti da na papir učenika 0 napiše [10], na papir učenika 3 napiše [1, 63, 4], a papir učenika 2 ostavi prazan. Da bi se to postiglo, procedura mora vratiti $B = [[10], [], [], [1, 63, 4]]$.

Nakon što nastavnik 0 ode, učenici razmjenjuju svoje papire prema P . Nakon razmjene, papiri su $A = [[10], [], [1, 63, 4], []]$.

Ocjenjivač sada poziva:

```
process_step(4, 2, 1, [0, 3], [[10], [], [1, 63, 4], []])
```

Učenici 0 i 3 su podigli ruku, tako da se njihovi radovi ne mogu mijenjati. Nastavnik 1 može odlučiti da na papir učenika 1 napiše [0, 40], a na papir učenika 2 [1, 50]. Da bi to uradila, procedura mora vratiti $B = [[10], [0, 40], [1, 50], []]$.

Nakon što nastavnik 1 ode, papiri se ponovo zamjenjuju prema P . Nakon zamjene, papiri su $A = [[10], [1, 50], [], [0, 40]]$.

Konačno, ocjenjivač proglašava:

```
determine_steps(4, 2, [[10], [1, 50], [], [0, 40]])
```

Ova procedura mora vratiti niz $D = [1, 0, -1, 1]$ jer su studenti 0 i 3 digli ruku u koraku 1, student 1 je digao ruku u koraku 0, a student 2 nikada nije digao ruku. U ovom primjeru, $C = 3$.

Primjer ocjenjivača

Format unosa:

```
NM
Q[0] Q[1] ... Q[N-1]
P[0] P[1] ... P[N-1]
```

Ovdje je Q niz dužine N , gdje je $Q[i] = j$ ako učenik i podigne ruku tokom koraka j , ili $Q[i] = -1$ ako učenik i nikada ne podigne ruku tokom cijele igre.

Izlazni format:

Nakon svakog poziva funkcije `process_step`, ocjenjivač uzorka ispisuje sadržaj radova. Neka je K dužina od B i $L[i]$ dužina od $B[i]$ (za svako $0 \leq i < K$). Primjer ocjenjivača ispisuje B u sljedećem formatu, **nakon čega slijedi prazan red**:

```
K
L[0] B[0][0] B[0][1] ... B[0][L[0]-1]
L[1] B[1][0] B[1][1] ... B[1][L[1]-1]
:
L[K-1] B[K-1][0] B[K-1][1] ... B[K-1][L[K-1]-1]
```

Ocjenjivač uzorka zatim zamjenjuje papire prema permutaciji P . Ako je $K \neq N$, ocjenjivač uzorka ispisuje poruku o grešci i umjesto toga se završava.

Nakon pozivanja funkcije `determine_steps`, primjer ocjenjivača ispisuje:

```
CH
D[0] D[1] ... D[H-1]
```

Ovdje je H dužina niza D kojeg vraća funkcija `determine_steps`.