

Sinifdə Oyun

Daşkənddəki tanınmış bir orta məktəb ildönümünü şagirdlər və müəllimlər arasında oyun keçirməklə qeyd edir. Sinifdə 0 -dan $N - 1$ -ə qədər nömrələnmiş N şagird oturub. Hər şagirdin əlində bir kağız var və kağızın üzərində tam ədədlərdən ibarət massiv var. Əvvəlcə hər kağız boşdur, yəni kağız boş bir massivdən ibarətdir.

Şagirdlər oyunu M müəllim və 1 direktorla oynayır. Müəllimlər 0-dan $M - 1$ -ə qədər nömrələnib. Oyun M addımdan ibarətdir və addımlar da 0-dan $M - 1$ -ə qədər nömrələnib. j addımında j -ci müəllim sinfə daxil olur və aşağıdakı hadisələr baş verir:

- Şagirdlərdən bəziləri (0 da ola bilər) əl qaldırır. Hər addımda **ən azı bir** şagirdin əlini **qaldırmadığına** və hər şagirdin bütün oyun ərzində əlini **ən çoxu bir dəfə** qaldıra biləcəyinə zəmanət verilir.
- Müəllim hazırda şagirdlərin əlində olan kağızlara baxıb, bu addımda əl qaldırmayan şagirdlərin kağızlarındakı massivi **dəyişdirə** bilər. Hər bir belə şagird üçün müəllim kağızın məzmununu yeni (boş da ola bilər) tam ədədlər massivi ilə əvəz edə bilər. Hər tam ədəd 0 ilə 63 arasında olmalıdır və massivdə maksimum 63 element ola bilər.
- Bu dəyişikliklərdən sonra j -ci müəllim dərstdən çıxır və şagirdlər gizli P permutasiyasına uyğun olaraq kağızlarını dəyişirlər. Yəni, hər bir i ($0 \leq i < N$) şagirdi işini $P[i]$ şagirdinə verir, burada $P[0], P[1], \dots, P[N - 1]$ ədədləri, 0 və $N - 1$ arasında olan **fərqli** tam ədədlərdir. P -nin dəyərləri oyunun bütün mərhələlərində **sabit** olur, lakin müəllimlər və direktor onları bilmir.

M addımın hamısı başa çatdıqdan və P -yə uyğun son kağız dəyişməsi aparıldıqdan sonra direktor daxil olur. Direktor yalnız şagirdlərin əlində olan kağızlara baxaraq, hər bir i ($0 \leq i < N$) şagirdi üçün i -in əlini qaldırdığı j -ci addımı müəyyən etməli və ya i -in heç vaxt əlini qaldırmadığı qənaətinə gəlməlidir.

Müəllimlər kağızlarda yazılmış tam ədədlər massivlərindən başqa vasitələr ilə digər müəllimlərlə və ya direktorla əlaqə saxlaya bilmirlər. Hər bir müəllim sinfə hansı addımda girdiyini bilir.

Tapşırığınız müəllimlər və direktor üçün hər bir şagirdin əlini nə vaxt qaldırıb-qaldırmadığını düzgün müəyyən edən bir strategiya hazırlamaq və həyata keçirməkdir. Balınız kağız üzərində yazılmış istənilən massivin maksimum uzunluğundan asılı olacaq: daha qısa maksimum uzunluq daha yüksək və ya bərabər bal verə bilər.

İcra Detalları

İki prosedur icra etməlisiniz, biri müəllimlər, digəri isə direktor üçün.

Müəllimlər üçün icra etməli olduğunuz prosedur belədir:

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

- N : şagirdlərin sayı.
- M : addımların sayı və müəllimlərin sayı.
- R : cari addım nömrəsi (0 -dan $M - 1$ -a qədər).
- T : bu addımda əlini qaldıran şagirdlərin indekslərindən ibarət (boş ola bilən) bir massiv.
- A : kağızları təsvir edən N uzunluğunda bir massiv, burada $A[i]$ ($0 \leq i < N$) addımın əvvəlində i -ci şagird tərəfindən saxlanılan kağızdakı tam ədədlər massividir.
- Bu prosedur hər oyun üçün tam olaraq M dəfə, $R = 0, 1, \dots, M - 1$ sırası ilə çağırılır.

Prosedur, müəllimin dəyişikliklərindən sonra kağızları qaytarır, A ilə eyni formatda, N uzunluğunda B massivi qaytarmalıdır.

- Əlini qaldıran hər i şagirdi üçün (yəni, i indeksi T massivindədir), kağız dəyişdirilməməlidir, beləliklə, $B[i] = A[i]$.
- $0 \leq i < N$ olan hər bir i üçün, $B[i]$ -in uzunluğu 63 keçməməlidir və $B[i]$ -dəki hər tam ədəd 0 ilə 63 arasında olmalıdır.

Direktor üçün icra etməli olduğunuz prosedur belədir:

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

- N, M : yuxarıdakı ilə eynidir.
- A : N uzunluğunda bir massiv, burada $A[i]$, M addımdan sonra şagirdin i -də saxladığı kağızdakı tam ədədlər massividir.
- Bu prosedur `process_step` funksiyasının son çağırışından sonra, hər oyunda bir dəfə çağırılır.

Prosedur, N uzunluğunda D massivi qaytarmalıdır. $0 \leq i < N$ olan hər i üçün, D massivi, bunu saxlamalıdır:

- Əgər i şagirdi j addımında əlini qaldırıbsa $D[i] = j$, və ya
- Əgər i şagirdi oyun boyu əlini qaldırmayıbsa, $D[i] = -1$.

Proqramınızın `process_step` qaytarma dəyəri istisna olmaqla, bir `process_step` çağırışından digərinə heç bir məlumatın saxlanmasına və ya ötürülməsinə icazə verilmir. Proqramınız bunu etməyə cəhd edərsə, CMS-dəki balınız səhv ola bilər və yarışma bitdikdən sonra azala bilər. Nəzərə alın ki, qiymətləndirici proqramınızın birdən çox nüsxəsini eyni vaxtda icra edə bilər. Bu o deməkdir ki, eyni oyundan `process_step` və `determine_steps` çağırışları fərqli nüsxədə, fərqli oyunlardan `process_step` və `determine_steps` çağırışları isə eyni nüsxədə icra edilə bilər. Hər bir testə 5 oyun daxildir.

Məhdudiyyətlər

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Hər bir şagird oyun ərzində **ən çoxu bir dəfə** əlini qaldırır. Başqa sözlə, hər bir i şagirdi üçün i əlini qaldırdığı ən çoxu bir j addımı var.
- Hər addımda ən azı bir şagird əlini **qaldırmır**.

Alt tapşırıqlar və Qiymətləndirmə

Alt tapşırıq	Bal	Əlavə Məhdudiyyətlər
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ hər bir $0 \leq i \leq N - 1$.
4	25	Hər addımda maksimum bir şagird əlini qaldırır.
5	56	Əlavə məhdudiyyət yoxdur.

Hər bir test üçün, əgər hər hansı bir `process_step` çağırışının qaytardığı dəyər etibarsızdırsa və ya hər hansı bir `determine_steps` çağırışının qaytardığı dəyər səhvdirsə, balınız 0 -dir (CMS-də `Output isn't correct` kimi göstərilir).

Əks halda, $C_{\text{process_step}}$ ilə qaytarılan **istənilən** B massivindəki **istənilən** massivin maksimum uzunluğu olsun. Daha sonra, S balı olan alt tapşırıqdakı testin balı $S \cdot X$ kimi hesablanır, burada X aşağıdakı cədvələ əsasən C -dən asılıdır:

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

Nümunə

$N = 4$ şagird, $M = 2$ müəllim və $P = [0, 3, 1, 2]$ permutasiyasının olduğu bir ssenarini nəzərdən keçirin. Əvvəlcə bütün kağızlar boşdur.

Qiymətləndirici ilk öncə çağırır:

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

1-ci şagird əlini qaldırdığı üçün onun kağızı dəyişdirilə bilməz. 0-cı müəllim 0-cı şagirdin kağızına [10] yazmaq, 3-cü şagirdin kağızına [1, 63, 4] yazmaq və 2-ci şagirdin kağızını boş qoymaq qərarına gələ bilər. Bunu etmək üçün prosedur $B = [[10], [], [], [1, 63, 4]]$ qaytarmalıdır.

0-cı müəllim getdikdən sonra şagirdlər kağızlarını P -yə uyğun olaraq dəyişdirirlər. Dəyişiklikdən sonra kağızlar $A = [[10], [], [1, 63, 4], []]$ olur.

Qiymətləndirici indi çağırır:

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

0-cı və 3-cü tələbələr əllərini qaldırdıqları üçün kağızları dəyişdirilə bilməz. 1-ci müəllim 1-ci tələbənin kağızına [0, 40], 2-ci tələbənin kağızına isə [1, 50] yazmaq qərarına gələ bilər. Bunu etmək üçün prosedur $B = [[10], [0, 40], [1, 50], []]$ qaytarmalıdır.

1-ci Müəllim getdikdən sonra, kağızlar yenidən P -yə uyğun olaraq dəyişdirilir. Dəyişiklikdən sonra kağızlar $A = [[10], [1, 50], [], [0, 40]]$ olur.

Nəhayət, qiymətləndirici çağırır:

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

Bu prosedur 0-cı və 3-cü tələbələr 1-ci addımda, 1-ci tələbə 0-cı addımda və 2-ci tələbə isə heç vaxt əllini qaldırmadığı üçün $D = [1, 0, -1, 1]$ qaytarmalıdır. Bu nümunədə, $C = 3$.

Nümunə Qiymətləndirici

Giriş formatı:

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

Burada, Q , N uzunluğunda bir massivdir və $Q[i] = j$ tələbəsi i addımı zamanı əlini qaldırsa, $Q[i] = j$, və ya i tələbəsi bütün oyun ərzində heç vaxt əlini qaldırmazsa $Q[i] = -1$ olur.

Çıxış formatı:

`process_step` funksiyasına hər çağırışdan sonra nümunə qiymətləndirici kağızların məzmununu çıxarır. Tutaq ki, K B -in uzunluğu, $L[i]$ isə $B[i]$ in uzunluğudur (hər $0 \leq i < K$ üçün). Nümunə qiymətləndirici B -ni aşağıdakı formatda çap edir, **sonra boş sətir** yazılır:

```
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]
```

Nümunə qiymətləndirici daha sonra kağızları P permutasiyasına uyğun olaraq dəyişdirir. Əgər $K \neq N$ olarsa, nümunə qiymətləndirici xəta mesajını çap edir və proqramı bitirir.

`determine_steps` çağırıldıqdan sonra, nümunə qiymətləndirici aşağıdakı nəticəni verir:

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

Burada, H `determine_steps` tərəfindən qaytarılan D massivin uzunluğudur.