

Classroom Game

Տաշքենդի հայտնի ավագ դպրոցներից մեկը նշում է իր տարեդարձը՝ կազմակերպելով խաղ աշակերտների և ուսուցիչների միջև: Դասարանում նստած են N աշակերտներ՝ համարակալված 0-ից մինչև $N - 1$: Յուրաքանչյուր աշակերտ իր մոտ պահում է մի թուղթ, որը պարունակում է ամբողջ թվերի զանգված: Սկզբում յուրաքանչյուր թուղթ դատարկ է, այսինքն պարունակում է դատարկ զանգված:

Աշակերտները խաղում են M ուսուցիչների՝ համարակալված 0-ից մինչև $M - 1$, և դպրոցի տնօրենի հետ: Խաղն անցկացվում է M քայլով, որոնք նույնպես համարակալված են 0-ից մինչև $M - 1$: j -րդ քայլում j ուսուցիչը մտնում է դասարան, և տեղի են ունենում հետևյալ իրադարձությունները.

- Որոշ աշակերտներ (հնարավոր է՝ ոչ մեկը) ձեռք են բարձրացնում: Երաշխավորվում է, որ յուրաքանչյուր քայլում առնվազն մեկ աշակերտ **չի** բարձրացնում ձեռքը, և յուրաքանչյուր աշակերտ ամբողջ խաղի ընթացքում կարող է ձեռք բարձրացնել **առավելագույնը մեկ անգամ**:
- Ուսուցիչը նայում է աշակերտներից յուրաքանչյուրի թղթերին, և կարող է **փոփոխել** այն աշակերտների թղթերը, որոնք այս քայլում **չեն** բարձրացրել ձեռքը: Յուրաքանչյուր այդպիսի աշակերտի համար ուսուցիչը կարող է նրա թղթի պարունակությունը փոխարինել ամբողջ թվերի նոր (հնարավոր է՝ դատարկ) զանգվածով: Յուրաքանչյուր ամբողջ թիվ պետք է լինի 0-ից մինչև 63 ներառյալ, և զանգվածը կարող է ունենալ **առավելագույնը 63 տարր**:
- Այս փոփոխություններից հետո j համարի ուսուցիչը հեռանում է, և աշակերտները փոխանակում են իրենց թղթերը՝ համաձայն գաղտնի P տեղափոխության: Այսինքն՝ յուրաքանչյուր i աշակերտ ($0 \leq i < N$) իր թուղթը տալիս է $P[i]$ աշակերտին, որտեղ $P[0], P[1], \dots, P[N - 1]$ -ը 0-ից մինչև $N - 1$ ներառյալ միջակայքում գտնվող N **տարբեր** ամբողջ թվեր են: P -ի արժեքները խաղի բոլոր քայլերի ընթացքում **անփոփոխ** են, սակայն ուսուցիչներն ու տնօրենը դրանք չգիտեն:

Երբ բոլոր M քայլերն ավարտվել են և կատարվել է նաև վերջին թղթերը իրար փոխանակելու քայլը (համաձայն P -ի), դասարան է մտնում տնօրենը: Նայելով միայն աշակերտների մոտ գտնվող թղթերին՝ տնօրենը յուրաքանչյուր i աշակերտի համար ($0 \leq i < N$) պետք է պարզի այն j քայլի համարը, որի ընթացքում i աշակերտը բարձրացրել է ձեռքը, կամ եզրակացնի, որ i աշակերտը երբեք ձեռք չի բարձրացրել:

Ուսուցիչները չեն կարող հաղորդակցվել մյուս ուսուցիչների կամ տնօրենի հետ, բացառությամբ այն ամբողջ թվերի զանգվածների միջոցով, որոնք գրված են

աշակերտների միմյանց փոխանցվող թղթերի վրա: Յուրաքանչյուր ուսուցիչ գիտի քայլի համարը, որի ժամանակ ինքը մտել է դասարան:

Ձեր խնդիրն է մշակել և իրականացնել ուսուցիչների և տնօրենի համար այնպիսի ռազմավարություն, որը ճիշտ որոշում է՝ արդյոք և երբ է յուրաքանչյուր աշակերտ ձեռք բարձրացրել: Ձեր միավորը կախված կլինի որևէ թղթի որևէ պահի գրված զանգվածի առավելագույն երկարությունից. որքան փոքր լինի այդ երկարությունը, այնքան բարձր կլինի ձեր միավորը:

Իրականացման մանրամասներ

Դուք պետք է իրականացնեք երկու ֆունկցիա՝ մեկը ուսուցիչների, մյուսը՝ տնօրենի համար:

Ուսուցիչների համար պետք է իրականացնեք հետևյալ ֆունկցիան.

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

- N : աշակերտների քանակը:
- M : քայլերի քանակը, ինչպես նաև ուսուցիչների քանակը:
- R : ընթացիկ քայլի համարը (0 -ից մինչև $M - 1$):
- T : (հնարավոր է՝ դատարկ) զանգված, որը բաղկացած է այս քայլում ձեռք բարձրացնող աշակերտների համարներից, սորտավորված աճման կարգով.
- A : N երկարությամբ զանգված, որը նկարագրում է թղթերը, որտեղ $A[i]$ -ն ($0 \leq i < N$) այն ամբողջ թվերի զանգվածն է, որը քայլի սկզբում գտնվում է i աշակերտի մոտ եղած թղթի վրա:
- Այս ֆունկցիան յուրաքանչյուր խաղի ընթացքում կանչվում է ճիշտ M անգամ՝ $R = 0, 1, \dots, M - 1$ հերթականությամբ:

Ֆունկցիան պետք է վերադարձնի N երկարությամբ B զանգված, որը ներկայացնում է ուսուցչի փոփոխություններից հետո ստացված թղթերը՝ A -ի նույն ձևաչափով:

- Յուրաքանչյուր i աշակերտի համար, որը բարձրացրել է ձեռքը (այսինքն՝ i -ն հանդիպում է T -ում), թուղթը չպետք է փոխվի, հետևաբար $B[i] = A[i]$:
- Յուրաքանչյուր i -ի համար, որի համար $0 \leq i < N$, $B[i]$ -ի երկարությունը չպետք է գերազանցի 63-ը, և $B[i]$ -ի յուրաքանչյուր ամբողջ թիվ պետք է լինի 0 -ից մինչև 63 ներառյալ:

Տնօրենի համար պետք է իրականացնեք հետևյալ ֆունկցիան.

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

- N, M : Նույնն են, ինչ վերևում:
- A : N երկարությամբ զանգված, որտեղ $A[i]$ -ն այն ամբողջ թվերի զանգվածն է, որը բոլոր M քայլերից հետո գտնվում է i համարով աշակերտի մոտ եղած թղթի վրա:
- Այս ֆունկցիան յուրաքանչյուր խաղի ընթացքում կանչվում է ճիշտ մեկ անգամ `process_step`-ի վերջին կանչից հետո:

Ֆունկցիան պետք է վերադարձնի N երկարությամբ D զանգված: Յուրաքանչյուր i -ի համար, որի համար $0 \leq i < N$, պետք է կատարվի հետևյալը.

- $D[i] = j$, եթե i աշակերտը ձեռք է բարձրացրել j քայլի ընթացքում, կամ
- $D[i] = -1$, եթե i աշակերտը ամբողջ խաղի ընթացքում երբեք ձեռք չի բարձրացրել:

Ձեր ծրագրին չի թույլատրվում `process_step`-ի մի կանչից մյուսը պահել կամ փոխանցել որևէ տեղեկություն՝ բացառությամբ `process_step`-ի վերադարձվող արժեքի միջոցով: Եթե ձեր ծրագիրը փորձի դա անել, CMS-ում ձեր միավորը կարող է սխալ լինել և կարող է նվազեցվել մրցույթի ավարտից հետո: Նկատի ունեցեք, որ գրեյդերը կարող է միաժամանակ գործարկել ձեր ծրագրի մի քանի օրինակ: Սա նշանակում է, որ նույն խաղի `process_step` և `determine_steps` կանչերը կարող են իրականացվել տարբեր օրինակներում, իսկ տարբեր խաղերի `process_step` և `determine_steps` կանչերը կարող են իրականացվել նույն օրինակում: Յուրաքանչյուր թեստային օրինակ ներառում է մինչև 5 խաղ:

Սահմանափակումներ

- $2 \leq N \leq 63$
- $1 \leq M \leq 63$
- Յուրաքանչյուր աշակերտ ամբողջ խաղի ընթացքում ձեռք է բարձրացնում **առավելագույնը մեկ անգամ**: Այլ կերպ ասած՝ յուրաքանչյուր i աշակերտի համար կա առավելագույնը մեկ j քայլ, որի ընթացքում i -ն ձեռք է բարձրացնում:
- Յուրաքանչյուր քայլում առնվազն մեկ աշակերտ **չի** բարձրացնում ձեռքը:

Ենթախնդիրներ և գնահատում

Ենթախնդիր	Միավոր	Լրացուցիչ սահմանափակումներ
1	4	$M = 1$
2	6	$N = 2$
3	9	$P[i] = i$ բոլոր $0 \leq i \leq N - 1$ արժեքների համար:
4	25	Յուրաքանչյուր քայլում առավելագույնը մեկ աշակերտ է ձեռք բարձրացնում:
5	56	Լրացուցիչ սահմանափակումներ չկան:

Յուրաքանչյուր թեստի համար ձեր միավորը 0 կլինի (CMS-ում ցուցադրվում է որպես `Output isn't correct`), եթե `process_step` ֆունկցիայի որևէ կանչի վերադարձվող արժեքն անվավեր է կամ `determine_steps` ֆունկցիայի որևէ կանչի վերադարձվող արժեքը սխալ է:

Հակառակ դեպքում, թող C -ն լինի `process_step`-ի վերադարձած ցանկացած B -ում եղած **ցանկացած** զանգվածի առավելագույն երկարությունը: Այդ դեպքում S միավոր ունեցող ենթախնդրի թեստային օրինակի միավորը հաշվարկվում է որպես $S \cdot X$, որտեղ X -ը կախված է C -ից՝ համաձայն հետևյալ աղյուսակի.

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

Օրինակ

Դիտարկենք մի իրավիճակ, որտեղ կա $N = 4$ աշակերտ, $M = 2$ ուսուցիչ, և $P = [0, 3, 1, 2]$ տեղափոխությունը: Սկզբում բոլոր թղթերը դատարկ են:

Գրեյդերը սկզբում կանչում է.

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

Աշակերտ 1-ը բարձրացրել է ձեռքը, հետևաբար նրա թուղթը չի կարելի փոփոխել: Ուսուցիչ 0-ն կարող է որոշել $[10]$ գրել աշակերտ 0-ի թղթի վրա, $[1, 63, 4]$ գրել աշակերտ 3-ի թղթի վրա և աշակերտ 2-ի թուղթը թողնել դատարկ: Դա անելու համար ֆունկցիան պետք է վերադարձնի $B = [[10], [], [1, 63, 4]]$:

Ուսուցիչ 0-ի հեռանալուց հետո աշակերտները փոխանակում են իրենց թղթերը՝ համաձայն P -ի: Փոխանակումից հետո թղթերն են $A = [[10], [], [1, 63, 4], []]$:

Այժմ գրեյդերը կանչում է.

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

Աշակերտներ 0-ն և 3-ը բարձրացրել են ձեռքը, հետևաբար նրանց թղթերը չի կարելի փոփոխել: Ուսուցիչ 1-ը կարող է որոշել $[0, 40]$ գրել աշակերտ 1-ի թղթի վրա և $[1, 50]$ գրել աշակերտ 2-ի թղթի վրա: Դա անելու համար ֆունկցիան պետք է վերադարձնի $B = [[10], [0, 40], [1, 50], []]$:

Ուսուցիչ 1-ի հեռանալուց հետո թղթերը կրկին փոխանակվում են՝ համաձայն P -ի: Փոխանակումից հետո թղթերն են $A = [[10], [1, 50], [], [0, 40]]$:

Վերջապես, գրեյդերը կանչում է.

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

Այս ֆունկցիան պետք է վերադարձնի $D = [1, 0, -1, 1]$ զանգվածը, քանի որ աշակերտներ 0-ն և 3-ը ձեռք են բարձրացրել 1 քայլում, աշակերտ 1-ը ձեռք է բարձրացրել 0 քայլում, իսկ աշակերտ 2-ը երբեք ձեռք չի բարձրացրել: Այս օրինակում $C = 3$:

Նմուշային գրեյդեր

Մուտքային տվյալների ձևաչափը

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

Այստեղ Q -ն N երկարությամբ զանգված է, որտեղ $Q[i] = j$, եթե i աշակերտը ձեռք է բարձրացնում j քայլի ընթացքում, կամ $Q[i] = -1$, եթե i աշակերտը ամբողջ խաղի ընթացքում երբեք ձեռք չի բարձրացնում:

Ելքային տվյալների ձևաչափը

`process_step`-ի յուրաքանչյուր կանչից հետո նմուշային գերյոները արտածում է թղթերի պարունակությունները: Թող K -ն լինի B -ի երկարությունը, իսկ $L[i]$ -ը լինի $B[i]$ -ի երկարությունը (յուրաքանչյուր $0 \leq i < K$ -ի համար): Նմուշային գերյոները B -ն արտածում է հետևյալ ձևաչափով՝ **որից հետո գալիս է դատարկ տող**.

```
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]
```

Այնուհետև նմուշային գերյոները փոխանակում է թղթերը՝ համաձայն P տեղափոխության: Եթե $K \neq N$, նմուշային գերյոները արտածում է սխալի հաղորդագրություն և ավարտում է աշխատանքը:

`determine_steps`-ը կանչելուց հետո նմուշային գերյոները արտածում է.

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

Այստեղ H -ը `determine_steps`-ի վերադարձած D զանգվածի երկարությունն է: