

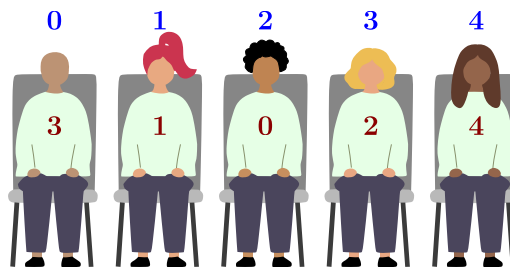
## D. Նստելու պլան (seatingplan)

EGOI-ի այս տարվա փակման արարողությանը մասնակցելու են  $N$  կարևոր հյուրեր: Նրանք բոլորը պետք է նստեն առաջին շարքում՝ դիվանագիտական արձանագրության բոլոր սրբություններին համապատասխան՝ շատ հստակ հերթականությամբ: Նոեմին երկու անգույն գիշեր է անցկացրել՝ նստելու ճիշտ կարգը պարզելու համար:

Վերոնիկան դեկավարում է փակման արարողությունը: Նրա բազմաթիվ պարտականություններից մեկն այն է, որ համոզվի, որ առաջին շարքի աթոռների վրա առկա են ճիշտ անվանատախտակները: Կա միայն մեկ փոքր խնդիր՝ Նոեմին նրան երբեք չի հայտնել նստելու ճիշտ կարգը, և այժմ Նոեմին չկա: Բարեբախտաբար, Դորկա լուսանկարիչն ունի մի հավելված, որը կարող է օգտակար լինել:

Դորկան պետք է նախապատրաստեր իր տեսախցիկները, որպեսզի կարողանար առաջին շարքի հյուրերի հատուկ լուսանկարներ անել: Կարգավորումների համար նրան անհրաժեշտ էր իմանալ, թե որքան լայն կլինի յուրաքանչյուր լուսանկարը, ուստի Նոեմին նրա համար ստեղծել էր մի հավելված, որը արագ արտածում է նրան անհրաժեշտ տեղեկատվությունը: Վերոնիկան այժմ ցանկանում է օգտագործել այդ հավելվածը՝ նստելու ճիշտ դասավորությունը պարզելու համար:

$N$  կարևոր հյուրերը համարակալված են 0-ից  $N - 1$ : Առաջին շարքի աթոռները նույնպես համարակալված են 0-ից  $N - 1$ , ձախից աջ: Յուրաքանչյուր  $I$ -ի համար ( $0 \leq I \leq N - 1$ ), թող  $g_I$ -ն լինի այն հյուրը, որը պետք է նստի  $I$  աթոռին, իսկ  $s_I$ -ն՝ այն աթոռը, որտեղ պետք է նստի  $I$  հյուրը:



Նկար 1: Հինգ հյուրերից բաղկացած շարք: Այս շարքի համար՝  $g = [3, 1, 0, 2, 4]$  և  $s = [2, 1, 3, 0, 4]$ :

Հավելվածն աշխատում է հետևյալ կերպ.

- Դորկան մուտքագրում է երեք տարբեր հյուրերի  $I$ ,  $J$ ,  $K$  համարները:
- Հավելվածը նրան հայտնում է այն հյուրերի նվազագույն քանակը, որոնք կերևան, եթե ընտրված երեք հյուրերն էլ լինեն լուսանկարում: Ֆորմալ առումով, հավելվածը կցուցադրի  $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$  արժեքը:

Օրինակ, տես Նկար 1-ում պատկերված իրավիճակը.

- $I = 0$ ,  $J = 2$  և  $K = 4$  հյուրերը նստած են  $s_I = 2$ ,  $s_J = 3$  և  $s_K = 4$  աթոռներին: Եթե Դորկան ընտրի նրանց, հավելվածը կցուցադրի  $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$  արժեքը:

Այլ կերպ ասած, ամենանեղ լուսանկարը, որը պարունակում է 0, 2 և 4 հյուրերին, ունի հենց այս երեք հյուրին:

- $I = 0$ ,  $J = 4$  և  $K = 3$  հյուրերը նստած են  $s_I = 2$ ,  $s_J = 4$  և  $s_K = 0$  աթոռներին: Եթե Դորկան ընտրի նրանց, հավելվածը կցուցադրի  $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$  արժեքը:

Այլ կերպ ասած, լուսանկարը, որը պարունակում է տրված երեք հյուրերին, պետք է պարունակի բոլոր 5 հյուրերին:

Օգնիր Վերոնիկային պարզել նստելու ճիշտ կարգը՝ օգտագործելով Դորկայի հավելվածը: Ավելի կոնկրետ՝ քո ծրագիրը պետք է որոշի և արտածի  $g_0, g_1, \dots, g_{N-1}$  հաջորդականությունը: Միշտ կա ճիշտ երկու պատասխան (մեկը մյուսի հակառակն է), և դու կարող ես արտածել դրանցից ցանկացածը: Քո միավորը կախված կլինի նրանից, թե քանի հարցում կկատարի քո ծրագիրը հավելվածին:

## Իրականացում

⇒ Սա ինտերակտիվ խնդիր է: Քո ծրագիրը կօգտագործի ստանդարտ մուտքային և ելքային հոսքերը՝ ստորև նկարագրված ձևաչափով ստուգող համակարգի (grader) հետ հաղորդակցվելու համար:

Քո ծրագիրը պետք է սկսի՝ կարդալով մուտքային տվյալների մեկ տող, որը պարունակում է դրական ամբողջ թիվ՝  $T$  (հաջորդող թեստերի քանակը):

Յուրաքանչյուր թեստի համար քո ծրագիրը պետք է սկսի՝ կարդալով մուտքային տվյալների մեկ տող, որը պարունակում է դրական ամբողջ թիվ՝  $N$  (աթոռների քանակը, որը նաև հյուրերի քանակն է):

Հարցում կատարելու համար քո ծրագիրը պետք է արտածի «?  $I J K$ » ձևաչափի տող, որտեղ  $0 \leq I, J, K \leq N - 1$  են երեք տարբեր թվեր:

Հարցում կատարելուց հետո քո ծրագիրը պետք է կարդա մեկ տող, որը պարունակում է մեկ դրական ամբողջ թիվ՝ քո հարցման պատասխանը:

Նստելու ճիշտ կարգի մասին պատասխան տալու համար քո ծրագիրը պետք է արտածի «!  $g_0 \dots g_{N-1}$ » ձևաչափի տող:

Բոլոր  $T$  փորձարկման դեպքերը լուծելուց հետո քո ծրագիրը պետք է ավարտվի սովորական ձևով:

Ուշադրություն դարձրու, որ CMS-ում քո լուծումը փորձարկելու համար օգտագործվող պաշտոնական ստուգող համակարգը կարող է լինել **ադապտիվ**: Դա նշանակում է, որ որոշ փորձարկման դեպքերի համար հյուրերի տեղաբաշխումը նախապես որոշված չէ: Փոխարենը՝ ստուգող համակարգը կարող է որոշել, թե մնացած հնարավոր տեղաբաշխումներից որն օգտագործել՝ կախված քո ծրագրի կողմից արդեն կատարված հարցումներից:

**Հոսքի թարմացում (Flushing)** Եթե դու չես օգտագործում տրամադրված ձևանմուշները, համոզվիր, որ յուրաքանչյուր տող տպելուց հետո թարմացնում ես ստանդարտ ելքային հոսքը (flush), հակառակ դեպքում քո ծրագիրը կարող է գնահատվել որպես *Սխալ*: Python-ում դա ավտոմատ կերպով տեղի է ունենում, եթե տողերը կարդալու համար օգտագործում ես `input()`, իսկ տպելու համար կարող ես օգտագործել `print(..., flush=True)`՝ այն հարկադրաբար թարմացնելու համար: C++-ում `cout << endl;`-ը տպելուց բացի նաև թարմացնում է հոսքը, իսկ `printf`-ի դեպքում օգտագործիր `fflush(stdout)`:

## Սահմանափակումներ

- $1 \leq T \leq 10$ :
- $N$  լինելու է 5 (միայն օրինակի համար), 8, 40 կամ 2000:
- Յուրաքանչյուր թեստի համար դու կարող ես կատարել առավելագույնը 10000 հարցում:

## Գնահատում

Քո ծրագիրը կփորձարկվի մի քանի փորձարկման դեպքերի վրա, որոնք խմբավորված են ըստ ենթախնդիրների: Ենթախնդրի համար միավոր ստանալու համար դու պետք է ճիշտ լուծես դրա մեջ ներառված բոլոր թեստերը:

- **Ենթախնդիր 0** [ 0 միավոր]: Օրինակ ( $N = 5$ ):
- **Ենթախնդիր 1** [ 9 միավոր]:  $N = 8$ :
- **Ենթախնդիր 2** [11 միավոր]:  $N = 2000$ , և 0 ու 1 հյուրերը նստած են կողք կողքի:
- **Ենթախնդիր 3** [15 միավոր]:  $N = 40$ :
- **Ենթախնդիր 4** [65 միավոր]:  $N = 2000$ :

1-ին և 2-րդ ենթախնդիրների համար ցանկացած լուծում, որը ճիշտ է լուծում բոլոր թեստերը, կստանա բոլոր միավորները:

3-րդ և 4-րդ ենթախնդիրների համար քո լուծումը պետք է ճիշտ լուծի բոլոր թեստերը՝ որևէ միավոր ստանալու համար, և քո միավորը կախված կլինի  $Q_s$ -ից՝ այն հարցումների առավելագույն քանակից, որը քո լուծումը պահանջել է թեստը լուծելու համար: Դիցուք  $X_s = \max(1, Q_s/N)$ : 3-րդ և 4-րդ ենթախնդիրների միավորները հաշվարկվում են հետևյալ կերպ.

$$\text{score}_3 = \min\left(15, 3 + \frac{19}{X_s^{1.5}}\right), \quad \text{score}_4 = \min\left(65, 3 + \frac{91}{X_s^{1.5}}\right)$$

$\text{score}_s$ -ի արժեքը կլորացվում է մինչև ամենամոտ ամբողջ թիվը յուրաքանչյուր ենթախնդրի համար, իսկ քո ընդհանուր միավորը դրանց գումարն է: Ամբողջական միավոր ստանալու համար դու պետք է 3-րդ ենթախնդիրը լուծես առավելագույնը 55 հարցումով, իսկ 4-րդ ենթախնդիրը՝ առավելագույնը 2597 հարցումով: 3-րդ և 4-րդ ենթախնդիրների  $Q_s$ -ի և միավորների օրինակելի արժեքները ցույց են տրված ստորև:

$Q_s$	55	56	60	70	80	100	150	10000
$\text{score}_3$	15	14	13	11	10	8	6	3

$Q_s$	2597	2800	3000	4000	5000	6000	8000	10000
$\text{score}_4$	65	58	53	35	26	21	14	11

## Օրինակներ

Grader	Լուծում
1 5	
	? 0 2 4
3	
	? 3 0 1
3	
	? 0 4 3
5	
	! 3 1 0 2 4

### Բացատրություն

Օրինակ մուտքային տվյալները պարունակում են մեկ թեստ ( $T = 1$ )՝  $N = 5$  հյուրերով: Այս թեստում հյուրերի թաքնված դասավորությունը համապատասխանում է նկար 1-ին:

Օրինակ լուծման կողմից կատարված առաջին հարցումն է՝ 0, 2, 4: Այս հարցման 3 պատասխանը մեզ հայտնում է, որ այս հյուրերը նստած են, անհայտ հերթականությամբ, երեք հաջորդական աթոռներին՝ կողք կողքի:

Երկրորդ հարցման 3 պատասխանը մեզ նույնն է հայտնում 3, 0 և 1 հյուրերի մասին:

Այժմ կարող ենք եզրակացնել, որ 0 հյուրը պետք է նստի մեջտեղում, 2 և 4 հյուրերը՝ մի կողմում, իսկ 1 և 3 հյուրերը՝ մյուս կողմում:

Երրորդ հարցումից հետո մենք արդեն կարող ենք վստահ լինել, որ հյուրերը պետք է նստած լինեն կա՛մ [3, 1, 0, 2, 4] կարգով, կա՛մ հակառակ՝ [4, 2, 0, 1, 3] կարգով: Մենք կարող ենք արտածել դասավորություններից որևէ մեկը:

### Կոդի ձևանմուշներ և գնահատման մանրամասները CMS-ում

Մենք խստորեն խորհուրդ ենք տալիս օգտագործել տրամադրված C++ և Python կոդի ձևանմուշները: Դրանք ստուգում են, թե արդյոք ստուգող համակարգի հետ հաղորդակցությունը հաջող է եղել, և ճիշտ ավարտվում են, եթե ոչ:

Եթե դուք չեք օգտագործում տրամադրված ձևանմուշները, այն դեպքերում, երբ ձեր լուծումը սխալ է, CMS-ը կարող է ցուցադրել սխալ վճիռ: Օրինակ՝ «Output isn't correct»-ի փոխարեն կարող եք ստանալ «Execution killed by signal» կամ «Execution timed out (wall clock limit exceeded)»:

Մենք նաև խորհուրդ ենք տալիս օգտագործել թեստավորման գործիքը (տես ստորև)՝ ձեր լուծումը տեղայնորեն փորձարկելու համար մինչև այն ուղարկելը: Թեստավորման գործիքը ստուգում է ձեր լուծման արտածումները և հայտնում անվավեր հարցումների օգտագործման մասին:

### Թեստավորման գործիք

Ձեր լուծման փորձարկումը հեշտացնելու համար մենք տրամադրում ենք մի պարզ գործիք, որը կարող եք ներբեռնել CMS-ից: Գործիքի օգտագործումը կամընտիր է: Ուշադրություն

դարձրու, որ CMS-ի պաշտոնական ստուգող համակարգը տարբերվում է թեստավորման գործիքից:

Գործիքն օգտագործելու համար ձեզ անհրաժեշտ է մուտքային ֆայլ: Կարող եք օգտագործել տրամադրված օրինակելի `seatingplan.input0.txt` մուտքային ֆայլը կամ ստեղծել ձերը: Մուտքային ֆայլը պետք է սկսվի մի տողով, որը պարունակում է թեստերի  $T$  քանակը, այնուհետև յուրաքանչյուր թեստի համար պետք է լինի երկու տող՝ մեկ տող  $N$  թվով, և մեկ տող՝  $g_0, g_1, \dots, g_{N-1}$  թվերով:

Python ծրագրերի համար, ենթադրենք `seatingplan.py` (սովորաբար աշխատում է որպես `pypy3 seatingplan.py`), աշխատեցրու թեստավորման գործիքը հետևյալ կերպ.

```
python3 testing_tool.py pypy3 seatingplan.py < seatingplan.input0.txt
```

C++ ծրագրերի համար նախ կոմպիլացրու լուծումը.

```
g++ -DEVAL -std=gnu++20 -O2 -pipe -static -s -o seatingplan seatingplan.cpp
```

իսկ հետո աշխատեցրու թեստավորման գործիքը.

```
python3 testing_tool.py ./seatingplan < seatingplan.input0.txt
```