

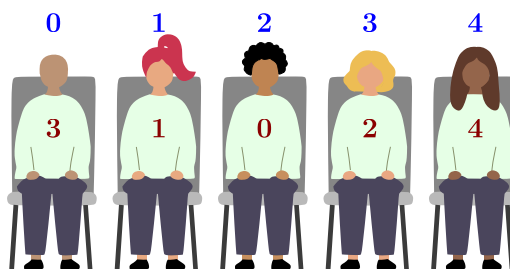
## D. Istumisplaan (seatingplan)

Selle EGOI lõputseremoonial on  $N$  väga olulist külalist. Nad peavad kõik istuma esireas väga spetsiifilises järjekorras, mis vastab diplomaatilise protokolliga kõikidele nüanssidele. Õige järjekorra leidmiseks kulus Noemil kaks unetut ööd.

Veronica vastutab lõputseremoonia eest. Üks tema paljudest ülesannetest on kindlaks teha, et esirea istmetel on kõigil õiged nimesildid. On vaid üks probleem: Noemi ei öelnud talle kunagi õiget järjekorda, ja nüüd on Noemi jäljetult kadunud. Õnneks on fotograaf Dorkal äpp, mis võib kasulikuks osutuda.

Dorka pidi oma kaamerad valmis panema selleks, et esirea külalistest paar väga spetsiifilist pilti teha. Valmistuseks oli tal vaja teada, kui lai iga pilt tuleb, seega tegi Noemi talle äpi, mis annab talle täpselt vajaliku informatsiooni. Veronica tahab nüüd seda äppi kasutada, et leida õige kohtade järjestus.

$N$  olulist külalist on nummerdatud 0 kuni  $N - 1$ . Esirea istekohad on samuti nummerdatud 0 kuni  $N - 1$  vasakult paremale. Iga  $I$  ( $0 \leq I \leq N - 1$ ) jaoks olgu  $g_I$  see, milline külaline peaks istuma kohal  $I$ , ja olgu  $s_I$  see, millisel istekohal peaks külaline  $I$  istuma.



Joonis 1: Rida viie külalisega. Selle rea jaoks  $g = [3, 1, 0, 2, 4]$  ja  $s = [2, 1, 3, 0, 4]$ .

Äpp töötab järgnevalt:

- Dorka sisestab arvud  $I$ ,  $J$  ja  $K$ , mis vastavad täpselt kolmele erinevale külalisele.
- Äpp ütleb talle minimaalse arvu külalisi, kes jäävad pildi peale, kui kõik kolm valitud külalist pildi peal on. Formaalsemalt näitab äpp väärtust  $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$ .

Näiteks vaatame olukorda, mis on kujutatud joonisel 1:

- Külalised  $I = 0$ ,  $J = 2$  ja  $K = 4$  on istekohtadel  $s_I = 2$ ,  $s_J = 3$  ja  $s_K = 4$ . Kui Dorka valib nemad, siis näitab äpp väärtust  $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$ .

Teiste sõnadega koosneb kõige kitsam foto, kus on peal külalised 0, 2 ja 4, täpselt nendest kolmest külalisest.

- Külalised  $I = 0$ ,  $J = 4$  ja  $K = 3$  on istekohtadel  $s_I = 2$ ,  $s_J = 4$  ja  $s_K = 0$ . Kui Dorka valib nemad, siis näitab äpp väärtust  $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$ .

Teiste sõnadega foto, kus on peal need kolm külalist, peab sisaldama kõiki 5 külalist.

Aita Veronical õige kohtade järjestus Dorka äppi kasutades kindlaks teha. Täpsemalt peaks sinu programm leidma ja väljastama jada  $g_0, g_1, \dots, g_{N-1}$ . Alati on täpselt kaks õiget vastust (millest üks on teise peegelpilt), neist võib väljastada ükskõik kumma. Sinu tulemus sõltub sellest, kui mitu päringut su lahendus teeb.

## Implementatsioon



See on interaktiivne ülesanne. Sinu lahendus kasutab standardsisendit ja standardväljundit hindamisprogrammiga suhtlemiseks, nii nagu allpool kirjeldatud.

Sinu programm peaks esiteks lugema sisse rea, kus on üks täisarv  $T$ : järgnevate testjuhtude arv.

Iga testjuhu jaoks peaks programm lugema sisendist ühe täisarvu  $N$ : toolide arv, mis on samas ka külaliste arv.

Päringu sooritamiseks peab programm väljastama rea kujul „?  $I\ J\ K$ “, kus  $0 \leq I, J, K \leq N - 1$  on kolm erinevat täisarvu.

Pärast päringu sooritamist peaks programm lugema sisendist ühe täisarvu, milleks on päringu vastus.

Selleks, et vastata leitud kohtade järjekord, tuleb väljastada rida kujul „!  $g_0 \dots g_{N-1}$ “.

Pärast kõigi  $T$  testjuhu lahendamist peaks programm oma töö normaalselt lõpetama.

Pane tähele, et CMS-is kasutatav ametlik hindamisprogramm võib olla **adaptiivne**. See tähendab, et mõnes testis ei ole külaliste permutatsioon ette ära määratud. Selle asemel võib hindamisprogramm valida, millist allesjäänud permutatsioonidest kasutada, sõltuvalt sinu programmi esitatud päringu-test.

**Puhvrite tühjendamine.** Kui sa ei kasuta ühte antud mallidest, siis kontrolli, et tühjendad standardväljundi puhvri pärast iga rea väljastamist. Vastasel juhul võib programm saada hindeks *Not correct*. Pythonis juhtub see automaatselt, kui kasutada sisendi lugemiseks funktsiooni `input()`. C++-is `cout << endl`; väljastab reavahetuse ja tühjendab puhvri; kui kasutad selle asemel `printf`-i, siis kasuta `fflush(stdout)`.

## Piirangud

- $1 \leq T \leq 10$ .
- $N$  väärtus on 5 (ainult näites), 8, 40, või 2000.
- Igas testjuhus võib teha ülimalt 10 000 päringut.

## Hindamine

Sinu programmi testitakse mitmel testil, mis on grupeeritud alamülesanneteks. Et alamülesande eest punkte saada, pead korrektselt lahendama kõik sellesse kuuluvad testid.

- **Alamülesanne 0 [ 0 punkti]**: Näide ( $N = 5$ ).
- **Alamülesanne 1 [ 9 punkti]**:  $N = 8$ .
- **Alamülesanne 2 [11 punkti]**:  $N = 2000$ , ja külalised 0 ja 1 istuvad kõrvuti.
- **Alamülesanne 3 [15 punkti]**:  $N = 40$ .
- **Alamülesanne 4 [65 punkti]**:  $N = 2000$ .

Alamülesannetes 1 ja 2 saab iga lahendus, mis korrektselt kõik testid lahendab, maksimumpunktid.

Alamülesannetes 3 ja 4 peab sinu lahendus punktide saamiseks kõik testid korrektselt lahendama, ja sinu saadud punktisumma sõltub väärtusest  $Q_s$ , mis on suurim kogus päringuid, mis su lahendusel kulus, et üks testjuht lahendada. Olgu  $X_s = \max(1, Q_s/N)$ . Punktisumma alamülesannete 3 ja 4 jaoks on siis:

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

Arv  $\text{score}_s$  ümardatakse siis iga alamülesande jaoks lähima täisarvuni, ja sinu kogupunktisumma on nende arvude summa.

Täispunktide saamiseks tuleb lahendada alamülesanne 3 ülimalt 55 päringuga ja alamülesanne 4 ülimalt 2597 päringuga.  $Q_s$  näidisväärtused ja näidisskoorid alamülesannete 3 ja 4 jaoks on toodud järgnevas tabelis:

$Q_s$	55	56	60	70	80	100	150	10000
score <sub>3</sub>	15	14	13	11	10	8	6	3

$Q_s$	2597	2800	3000	4000	5000	6000	8000	10000
score <sub>4</sub>	65	58	53	35	26	21	14	11

## Näited

Hindamisprogramm	Lahendus
1 5	
	? 0 2 4
3	
	? 3 0 1
3	
	? 0 4 3
5	
	! 3 1 0 2 4

## Selgitus

Esimeses sisendis on üks testjuht ( $T = 1$ ), kus on  $N = 5$  külalist. Külaliste (peidetud) õige järjekord selles testjuhus vastab joonisele 1.

Näidislahenduse esimene päring on 0, 2, 4. Selle vastus 3 ütleb meile, et need külalised istuvad kolmel järjestikusel kohal (mingis tundmatus järjekorras).

Teise päringu vastus 3 ütleb meile sama külaliste 3, 0 ja 1 kohta.

Nüüd saame järeldada, et külaline 0 peab istuma keskel, külalised 2 ja 4 temast ühel pool ja külalised 1 ja 3 teisel pool.

Pärast kolmandat päringut saame juba kindlad olla, et külalised istuvad kas järjekorras [3, 1, 0, 2, 4] või selle ümberpööratud variandis [4, 2, 0, 1, 3]. Võime väljastada neist ükskõik kumma.

## Koodimallid ja lahenduse hinded CMS-is

Antud C++ ja Pythoni koodimallide kasutamine on tugevalt soovituslik. Need mallid kontrollivad, et suhtlus hindamisprogrammiga oli edukas, ja lõpetavad normaalselt töö, kui see ei olnud edukas.

Kui sa ei kasuta antud koodimalli, siis võib CMS näidata vale tulemust (juhul, kui sinu lahendus ei ole korrektne). Näiteks võid tulemuse „Output isn’t correct“ asemel saada „Execution killed by signal“ või „Execution timed out (wall clock limit exceeded)“.

Soovitame lisaks kasutada testimistöööriista (vaata allpool), et oma lahendust enne selle esitamist lokaalselt katsetada. Testimistöööriist kontrollib sinu lahenduse väljundit ja raporteerib vigaste pärin-gute kasutamist.

## Testimistöõriist

Lahenduse testimise hõlbustamiseks on saadaval lihtne tööriist, mis on võimalik CMS-ist alla laadida. Selle tööriista kasutamine on valikuline. Pane tähele, et CMS-is kasutatav ametlik hindamisprogramm erineb testimistöõriistast.

Tööriista kasutamiseks on vaja sisendfaili. Võid kasutada antud näidisfaili `seatingplan.input0.txt` või enda sisendi ise luua. Sisendfaili esimesel real peaks olema testjuhtude arv  $T$ . Edasi peaks iga testjuhu jaoks olema kaks rida: üks rida arvuga  $N$  ja teine arvudega  $g_0, g_1, \dots, g_{N-1}$ .

Pythoni programmide jaoks, näiteks `seatingplan.py` (mida jooksutaksid tavaliselt kui `pypy3 seatingplan.py`) jookсутa testimistöõriista järgnevalt:

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

C++ programmide jaoks kõigepealt kompileeri oma lahendus:

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

ja seejärel jookсутa testimistöõriista:

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