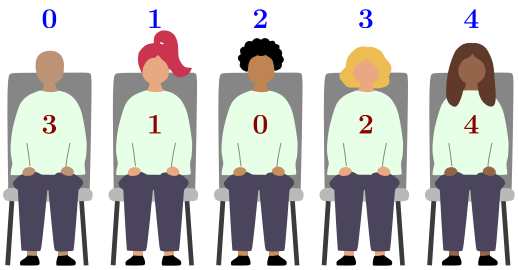
D. Stolplassering (seatingplan)

På avslutningsseremonien for årets EGOI skal N viktige gjester delta. Alle disse må sitte på første rad i en helt bestemt rekkefølge som tar hensyn til alle nyansene i den diplomatiske protokollen. Å bestemme den riktige rekkefølgen kostet Noemi to søvnløse netter.

Veronica har ansvaret for avslutningsseremonien. Et av hennes mange ansvarsområder er å sørge for at stolene på første rad har riktige navneskilt. Det er bare ett lite problem: Noemi fortalte henne aldri den riktige rekkefølgen på sitteplassene, og nå er Noemi sporløst forsvunnet. Heldigvis har fotografen Dorka en app som kan være nyttig.

Dorka måtte forberede kameraene sine slik at hun kunne ta noen spesifikke bilder av gjestene på første rad. For å sette opp utstyret trengte hun å vite hvor bredt hvert bilde kom til å bli, så Noemi lagde en app til henne som raskt gir henne informasjonen hun trenger. Veronica vil nå bruke appen for å finne ut av hvem som skal sitte hvor.

De N viktige gjestene er nummerert fra 0 til $N-1$. Stolene på første rad er også nummerert fra 0 til $N-1$, fra venstre til høyre. For hver I ($0 \leq I \leq N-1$), la g_I være gjesten som skal sitte på stol I, og la s_I være stolen hvor gjest I skal sitte.



Figur 1: En rad med fem gjester. For denne raden er $g = [3, 1, 0, 2, 4]$ og $s = [2, 1, 3, 0, 4]$.

Appen fungerer slik:

- Dorka skriver inn numrene I, J, K til nøyaktig tre forskjellige gjester.
- Appen forteller henne det minste antallet gjester som vil være synlige hvis alle de tre valgte gjestene er med på bildet. Formelt sett vil appen vise verdien $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$.

For eksempel, se situasjonen vist i Figur 1:

- Gjest $I = 0$, $J = 2$ og $K = 4$ sitter på stol $s_I = 2$, $s_J = 3$ og $s_K = 4$. Hvis Dorka velger dem, vil appen vise verdien $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$.

  Med andre ord har det smaleste bildet som inneholder gjest 0, 2 og 4 bare disse tre gjestene.

- Gjest $I = 0$, $J = 4$ og $K = 3$ er på stol $s_I = 2$, $s_J = 4$ og $s_K = 0$. Hvis Dorka velger dem, vil appen vise verdien $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$.

  Med andre ord må et bilde som inneholder disse tre gjestene inneholde alle 5 gjestene.

Hjelp Veronica med å finne riktig rekkefølge på gjestene ved hjelp av Dorkas app. Mer spesifikt skal programmet ditt finne og skrive ut sekvensen $g_0, g_1, ..., g_{N-1}$. Det finnes alltid nøyaktig to riktige svar (hvor det ene er det andre svaret baklengs), og du kan skrive ut hvilket som helst av dem. Poengsummen din vil avhenge av antallet spørringer til appen programmet ditt gjør.

Interaksjon

⇨ Dette er en interaktiv oppgave. Programmet ditt vil bruke standard input og output til å kommunisere med en grader på formatet beskrevet nedenfor.

Programmet ditt skal starte med å lese én linje fra input som inneholder et positivt heltall T, antallet tester som følger.

For hver test skal programmet ditt starte med å lese én linje fra input som inneholder et positivt heltall N, antall stoler, som også er antallet gjester.

For å gjøre en spørring, skal programmet ditt skrive ut en linje på formatet `? I J K`, hvor $0 \leq I, J, K \leq N-1$ er tre **forskjellige** tall.

Etter å ha gjort en spørring skal programmet ditt lese én linje som inneholder ett positivt heltall, svaret på spørringen din.

For å svare med en riktig rekkefølge, skal programmet ditt skrive ut en linje på formatet `! g_0 … g_{N-1}`.

Etter å ha løst alle de T testene skal programmet ditt avslutte på vanlig måte.

Merk at den offisielle graderen som brukes i CMS for å teste programmet ditt, kan være **adaptiv**. Det vil si at for noen tester er ikke rekkefølgen på gjestene bestemt på forhånd. I stedet kan graderen bestemme hvilken av de gjenværende rekkefølgene den skal bruke avhengig av de spesifikke spørringene som allerede er stilt av programmet ditt.

Flushing. Hvis du ikke bruker de vedlagte malene (templates), må du sørge for å flushe standard output etter å ha skrevet ut hver linje, ellers kan programmet ditt få *Not correct*. I Python skjer dette automatisk hvis du bruker `input()` for å lese linjer, og du kan bruke `print(..., flush=True)` for å tvinge frem en flush. I C++ flusher `cout << endl;` i tillegg til å skrive ut en ny linje; bruker du printf, bruk `fflush(stdout)`.

Begrensninger

- $1 \leq T \leq 10$.
- N vil være 5 (bare for eksempelet), 8, 40, eller 2000.
- For hver test kan du gjøre maksimalt 10 000 spørringer.

Poengsum

Programmet ditt vil bli testet på et sett med deloppgaver (subtasks). Hver deloppgave inneholder et sett med tester. For å få poengene for en deloppgave, må du løse alle testene i deloppgaven.

- **Deloppgave 0 [0 poeng]**: Eksempel ($N = 5$).
- **Deloppgave 1 [9 poeng]**: $N = 8$.
- **Deloppgave 2 [11 poeng]**: $N = 2000$, og gjest 0 og 1 sitter ved siden av hverandre.
- **Deloppgave 3 [15 poeng]**: $N = 40$.
- **Deloppgave 4 [65 poeng]**: $N = 2000$.

For deloppgave 1 og 2 vil ethvert program som løser alle testene riktig få alle poengene.

For deloppgave 3 og 4 må programmet ditt løse alle testene riktig for å få poeng, og poengsummen din vil avhenge av Q_s, det høyeste antallet spørringer programmet ditt trengte for å løse en test. La $X_s = \max(1, Q_s/N)$. Poengene for deloppgave 3 og 4 regnes da ut slik:

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

Verdien av score_s blir rundet av til nærmeste heltall for hver deloppgave, og totalscoren din er summen av disse. For å få full poengsum må du løse deloppgave 3 med maksimalt 55 spørringer og deloppgave 4 med maksimalt 2597 spørringer. Eksempler på verdier av Q_s og poeng for deloppgave 3 og 4 er vist nedenfor.

Q_s	55	56	60	70	80	100	150	10000
score_3	15	14	13	11	10	8	6	3

Q_s	2597	2800	3000	4000	5000	6000	8000	10000
score_4	65	58	53	35	26	21	14	11

Eksempler

Grader	Din løsning
1 5	
	? 0 2 4
3	
	? 3 0 1
3	
	? 0 4 3
5	
	! 3 1 0 2 4

Forklaring av eksempler

Eksempel-inputen inneholder én test ($T = 1$) med $N = 5$ gjester. Den skjulte plasseringen av gjestene i denne testen samsvarer med Figur 1.

Den første spørringen gjort av eksempelløsningen er 0, 2, 4. Svaret 3 på denne spørringen forteller oss at disse gjestene sitter på tre sammenhengende stoler ved siden av hverandre, i en eller annen ukjent rekkefølge.

Svaret 3 på den andre spørringen forteller oss det samme om gjest 3, 0 og 1.

Vi kan nå utlede at gjest 0 må sitte i midten, med gjest 2 og 4 på den ene siden og gjest 1 og 3 på den andre siden.

Etter den tredje spørringen kan vi allerede være sikre på at gjestene må sitte enten i rekkefølgen $[3, 1, 0, 2, 4]$ eller i motsatt rekkefølge $[4, 2, 0, 1, 3]$. Vi kan skrive ut hvilken som helst av disse.

Kodemaler og evalueringsdetaljer i CMS

Vi anbefaler på det sterkeste å bruke de vedlagte kodemalene for C++ og Python. Disse sjekker om kommunikasjonen med graderen var vellykket og avslutter på en ryddig måte hvis den ikke var det.

Hvis du ikke bruker de vedlagte malene, kan dette føre til at CMS viser feil bedømmelse, i tilfellet hvor løsningen din er feil. For eksempel, i stedet for "Output isn't correct", kan du få "Execution killed by signal" eller "Execution timed out (wall clock limit exceeded)".

Vi anbefaler også å bruke testverktøyet (se nedenfor) til å teste programmet ditt lokalt før du sender det inn. Testverktøyet sjekker output fra programmet ditt og rapporterer bruk av ugyldige spørringer.

Testverktøy

For å gjøre det enklere å teste programmet ditt, tilbyr vi et enkelt testverktøy du kan laste ned fra CMS. Det er valgfritt å bruke verktøyet. Merk at den offisielle graderen på CMS er annerledes enn testverktøyet.

For å bruke verktøyet trenger du en inputfil. Du kan bruke den vedlagte eksempel-inputen `seatingplan.input0.txt` eller lage din egen. Inputfilen skal starte med en linje som har antall tester T, og deretter skal den ha to linjer per test: én linje med tallet N og deretter én linje med tallene $g_0, g_1, ..., g_{N-1}$.

For Python-programmer, for eksempel `seatingplan.py` (normalt kjørt som `pypy3 seatingplan.py`), kjør testverktøyet slik:

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

For C++-programmer må du først kompilere programmet ditt:

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

og deretter kjøre testverktøyet:

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