

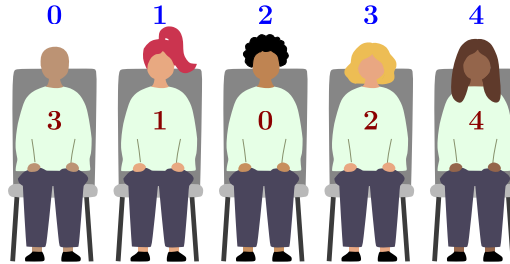
D. Oturma Planı (seatingplan)

Bu EGOI-nin bağlanmış mərasimində N vacib qonaq iştirak edəcək. Bütün bu qonaqlar diplomatik protokolun bütün nüanslarına uyğun olaraq çox spesifik bir sırayla ön cərgədə oturdulmalıdır. Doğru oturma sırasını müəyyən etmək Noemi-nin iki yuxusuz gecəsini aldı.

Veronika bağlanmış mərasiminə nəzarət edir. Onun çoxsaylı vəzifələrindən biri ön cərgədəki oturacaqlarda düzgün ad lövhələrinin olduğuna əmin olmaqdır. Yalnız kiçik bir problem var: Noemi ona heç vaxt düzgün oturma sırasını deməyib və indi Noemini heç yerdə tapmaq mümkün deyil. Xoşbəxtlikdən, fotoqraf Dorkanın işə yaraya biləcək bir tətbiqi var.

Dorka birinci cərgədəki qonaqların bəzi xüsusi şəkillərini çəkə bilmək üçün kameralarını hazırlamalı idi. Quraşdırma üçün o, hər bir şəklin nə qədər geniş olacağını bilməli idi, buna görə də Noemi ona ehtiyacı olan məlumatı tez bir zamanda verən bir tətbiq hazırladı. İndi Veronika düzgün oturma planını tapmaq üçün tətbiqdən istifadə etmək istəyir.

N vacib qonaq 0-dan $N - 1$ -ə qədər nömrələnib. Ön cərgədəki oturacaqlar da soldan sağa 0-dan $N - 1$ -ə qədər nömrələnib. Hər bir I üçün ($0 \leq I \leq N - 1$), g_I I oturacağında oturmalı olan qonağı, s_I isə I qonağının oturmalı olduğu oturacağı ifadə etsin.



Şəkil 1: Beş qonaqlı bir cərgə. Bu cərgə üçün $g = [3, 1, 0, 2, 4]$ və $s = [2, 1, 3, 0, 4]$.

Tətbiq aşağıdakı kimi işləyir:

- Dorka tam olaraq üç fərqli qonağın I , J , K nömrələrini daxil edir.
- Tətbiq, seçilmiş hər üç qonaq şəkində olarsa, görünəcək minimum qonaq sayını ona deyir. Formal olaraq, tətbiq $\max(s_I, s_J, s_K) - \min(s_I, s_J, s_K) + 1$ dəyərini göstərəcək.

Məsələn, Şəkil 1-də göstərilən vəziyyətə baxın:

- $I = 0$, $J = 2$ və $K = 4$ qonaqları $s_I = 2$, $s_J = 3$ və $s_K = 4$ oturacaqlarındadırlar. Əgər Dorka onları seçərsə, tətbiq $\max(2, 3, 4) - \min(2, 3, 4) + 1 = 3$ dəyərini göstərəcək.

Başqa sözlə, 0, 2 və 4 qonaqlarının olduğu ən dar şəkildə yalnız bu üç qonaq var.

- $I = 0$, $J = 4$ və $K = 3$ qonaqları $s_I = 2$, $s_J = 4$ və $s_K = 0$ oturacaqlarındadırlar. Əgər Dorka onları seçərsə, tətbiq $\max(2, 4, 0) - \min(2, 4, 0) + 1 = 5$ dəyərini göstərəcək.

Başqa sözlə, verilmiş üç qonağın olduğu şəkildə bütün 5 qonaq da olmalıdır.

Veronikaya Dorkanın tətbiqindən istifadə edərək düzgün oturma sırasını müəyyən etməyə kömək edin. Daha dəqiq desək, programınız g_0, g_1, \dots, g_{N-1} ardıcılığını müəyyən etməli və çap etməlidir. Həmişə tam olaraq iki doğru cavab var (biri digərinin tərsi olmaqla) və siz onlardan istənilən birini çap edə bilərsiniz. Sizin xalınız həllinizin tətbiq edəcəyi sorğuların sayından asılı olacaq.

İcra



Bu, interaktiv məsələdir. Proqramınız aşağıda təsvir olunan formatda qiymətləndirici ilə əlaqə qurmaq üçün standart giriş və çıxışdan istifadə edəcəkdir.

Proqramınız ardınca gələn testlərin sayı olan müsbət tam T ədədinin olduğu bir sətir oxumaqla başlamalıdır.

Hər bir test üçün, proqramınız həm qonaqların, həm də oturacaqların sayı olan müsbət tam N ədədinin olduğu bir sətir oxumaqla başlamalıdır.

Sorğu etmək üçün proqramınız “? I J K ” formatında bir sətir çap etməlidir, burada $0 \leq I, J, K \leq N - 1$ şərtini ödəyən üç **fərqli** ədəddir.

Sorğu etdikdən sonra proqramınız sorğunuzun cavabı olan bir müsbət tam ədəddən ibarət bir sətir oxumalıdır.

Düzgün oturma sırası ilə cavab vermək üçün proqramınız “! $g_0 \dots g_{N-1}$ ” formatında bir sətir çap etməlidir.

Bütün T testlərini həll etdikdən sonra proqramınız normal şəkildə sonlanmalıdır.

Nəzərə alın ki, həllinizi yoxlamaq üçün CMS-də istifadə olunan rəsmi qiymətləndirici **adaptiv** ola bilər. Yəni, bəzi testlər üçün qonaqların yerdəyişməsi əvvəlcədən müəyyən edilmir. Bunun əvəzinə, qiymətləndirici proqramınız tərəfindən artıq verilmiş sorğulardan asılı olaraq qalan yerdəyişmələrdən hansını istifadə edəcəyinə qərar verə bilər.

Flushing (Buferi təmizləmə). Əgər verilmiş şablonlardan istifadə etmirsinizsə, hər sətirdən sonra standart çıxışı flush etdiyinizə (buferi təmizlədiyinizə) əmin olun, əks halda proqramınız *Not correct* (Doğru deyil) kimi qiymətləndirilə bilər. Python-da, əgər sətirləri oxumaq üçün `input()` istifadə edirsinizsə, bu avtomatik baş verir. C++-da `cout << endl;` yeni sətir çap etməklə yanaşı həm də flush edir; əgər `printf` istifadə edirsinizsə, `fflush(stdout)` yazın.

Məhdudiyyətlər

- $1 \leq T \leq 10$.
- N 5 (yalnız nümunə), 8, 40 və ya 2000 olacaq.
- Hər test üçün ən çox 10000 sorğu edə bilərsiniz.

Qiymətləndirmə

Proqramınız alt tapşırıqlar şəklində qruplaşdırılmış bir neçə testlərdə yoxlanılacaq. Bir alt tapşırıq üçün xal qazanmaq üçün onun daxilindəki bütün testləri düzgün həll etməlisiniz.

- **Alt-tapşırıq 0** [**0 xal**]: Nümunə ($N = 5$).
- **Alt-tapşırıq 1** [**9 xal**]: $N = 8$.
- **Alt-tapşırıq 2** [**11 xal**]: $N = 2000$, və 0 ilə 1 qonaqları yan-yana otururlar.
- **Alt-tapşırıq 3** [**15 xal**]: $N = 40$.
- **Alt-tapşırıq 4** [**65 xal**]: $N = 2000$.

Alt tapşırıq 1 və 2 üçün bütün testləri düzgün həll edən istənilən həll bütün xallarla mükafatlandırılacaqdır.

Alt tapşırıq 3 və 4-də xal qazanmaq üçün həlliniz bütün testləri düzgün həll etməlidir və xalınız həllinizin bir testi həll etmək üçün etdiyi ən böyük sorğu sayı olan Q_s -dən asılı olacaq. Tutaq ki, $X_s = \max(1, Q_s/N)$. Alt tapşırıq 3 və 4 üçün xallar aşağıdakı kimi hesablanır:

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

$score_s$ dəyəri hər bir alt tapşırıq üçün ən yaxın tam ədədə yuvarlaqlaşdırılır və ümumi xalınız bunların cəmidir. Tam xal toplamaq üçün, alt tapşırıq 3-ü ən çox 55 sorğu ilə, alt tapşırıq 4-ü isə ən çox 2597 sorğu ilə həll etməlisiniz. Q_s -nin nümunə dəyərləri və alt tapşırıq 3 və 4 üçün xallar aşağıda göstərilmişdir.

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

Nümunələr

Qreyder	Həll
1 5	
	? 0 2 4
3	
	? 3 0 1
3	
	? 0 4 3
5	
	! 3 1 0 2 4

İzah

Nümunə giriş $N = 5$ qonaq olan bir testi ($T = 1$) ehtiva edir. Bu testdə qonaqların gizli konfigurasiyası Şəkil 1-ə uyğundur.

Nümunə həllinin etdiyi ilk sorğu 0, 2, 4-dür. Bu sorğunun cavabının 3 olması, bu qonaqların hər hansı naməlum ardıcılıqla ardıcıl üç oturacaqda yan-yanə oturduğunu bildirir.

İkinci sorğunun cavabının 3 olması, 3, 0 və 1 qonaqları üçün də eyni şeyi bildirir.

İndi biz belə bir nəticə çıxara bilərik ki, 0-cı qonaq ortada, 2 və 4-cü qonaqlar bir tərəfdə, 1 və 3-cü qonaqlar isə digər tərəfdə oturmalıdırlar.

Üçüncü sorğudan sonra biz artıq qonaqların ya [3, 1, 0, 2, 4] sırasıyla, ya da tərsinə [4, 2, 0, 1, 3] sırasıyla oturmalı olduğundan əmin ola bilərik. Biz istənilən sıranı çap edə bilərik.

CMS-də Kod Şablonları və Qiymətləndirmə Detalları

C++ və Python üçün təqdim edilən kod şablonlarından istifadə etməyinizi qətiyyətlə tövsiyə edirik. Bunlar qiymətləndirici ilə əlaqənin uğurlu olub-olmadığını yoxlayır və olmadıqda proqramı problemsiz sonlandırır.

Təqdim edilən şablonlardan istifadə etmirsənsə, həllinizin səhv olduğu hallarda CMS səhv hökm göstərə bilər. Məsələn, “Output isn’t correct” əvəzinə “Execution killed by signal” və ya “Execution timed out (wall clock limit exceeded)” görə bilərsiniz.

Həllinizi göndərməzdən əvvəl yerli olaraq test etmək üçün test alətindən (aşağıya baxın) istifadə etməyinizi də tövsiyə edirik. Test aləti həllinizin çıxışlarını yoxlayır və etibarsız sorğuların istifadəsini bildirir.

Test Aləti

Həllinizin yoxlanılmasını asanlaşdırmaq üçün CMS-dən yükləyə biləcəyiniz sadə bir alət təqdim edirik. Alətdən istifadə etmək məcburi deyil. Nəzərə alın ki, CMS-dəki rəsmi qiymətləndirici test alətindən fərqlidir.

Alətdən istifadə etmək üçün sizə bir giriş faylı lazımdır. Siz təqdim olunan nümunə giriş `seatingplan.input0.txt` faylından istifadə edə bilərsiniz və ya özünüzlükünü yarada bilərsiniz. Giriş faylı testlərin sayı olan T -nin olduğu bir sətirlə başlamalı və sonra hər test üçün iki sətir olmalıdır: birinci sətirdə N ədədi, daha sonra isə g_0, g_1, \dots, g_{N-1} ədədlərinin olduğu sətir.

Python proqramları üçün, məsələn, `seatingplan.py` (adətən `pypy3 seatingplan.py` kimi işə salınır) test alətini aşağıdakı kimi işə salın:

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

C++ proqramları üçün, əvvəlcə həllinizi kompilyasiya edin:

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

və sonra test alətini işə salın:

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