

D. Bağ Dekorasiyaları

Tapşırıq Adı	Garden Decorations
Zaman Limiti	7 saniyə
Yaddaş Limiti	1 GB

Hər gün məktəbə gedərkən və evə qayıdarkən Durna 0'dan $N - 1$ 'ə nömrələnmiş N sayda evin olduğu küçə boyunca gedir. Hazırda i -ci evdə i -ci adam yaşayır. Sakinlər mənzərə dəyişikliyi üçün bir-birləri ilə evlərini dəyişməyə qərar veriblər. i -ci evə köçəcək adam a_i -ci adamdır (hazırda a_i -ci evdə yaşayır).

Hər evin bağında quş heykəli var. Heykəllərin iki mümkün vəziyyəti var; qanadları *açıq* (sanki quş uçuş) ya da *qapalı* (sanki yerdə dayanır). Sakinlər onların quş heykəllərinin hansı vəziyyətdə olması mövzusunda çox seçicidirlər və hər biri iki vəziyyətdən birinə üstünlük verir. Hazırda i -ci evin qarşısındakı quş heykəli i -ci sakinin üstünlük verdiyi vəziyyətdədir. Sakinlər evin bağındakı heykəl onların istədikləri vəziyyətdə olmadıqca həmin evə köçməkdən imtina edirlər. Durna onlara köçə bilmələri üçün quş heykəllərini tənzimləməkdə kömək etmək istəyir.

Bunun üçün o, aşağıdakıları edir: hər dəfə küçə ilə getdikdə (məktəbə gedərkən və ya evə qayıdarkən), keçdiyi quş heykəllərini bir-bir müşahidə edir və ola bilsin bəzi heykəlləri tənzimləyir (qanadlarını açaraq və ya bağlayaraq). Məktəbdə və evdə günləri çox məşğul keçdiyindən **əvvəlki gəzintilərində gördüyü quş heykəllərinin vəziyyətlərini xatırlamır**. Xoşbəxtlikdən o, a_0, a_1, \dots, a_{N-1} siyahısını yazıb, ona görə də hansı sakinin hara köçdüyünü bilir.

Durnaya heykəlləri sakinlərin zövqünə uyğunlaşdırmaq üçün hansı quş heykəllərini manipulyasiya edəcəyini söyləyən strategiya hazırlamaqda kömək edin. O, küçə boyunca ən çox 60 dəfə gəzə bilər, lakin daha yüksək bal almaq üçün küçədə daha az sayda gəzməlidir.

İmplementasiya

Bu bir çox dəfə çalışdırılan məsələdir; yəni sizin proqramınız bir necə dəfə işləyəcəkdir.

Hər bir işləyişdə, gəzinin nömrəsini və evlərin sayını göstərən w və N iki tam ədədini oxumalısınız. Proqramınızın birinci işləyişində $w = 0$, ikincidə $w = 1$ və s. (daha detallı aşağıda izah edilmişdir).

Girişin ikinci sətirində, i -ci evə köçəcək olan və hal hazırda a_i -ci evdə yaşayan, a_0, a_1, \dots, a_{N-1} kimi sıralanmış N sayda tam ədəd verilir. a_i -lar permutasiya formalaşdırır, hansı ki, 0-dan $N - 1$ -ə kimi

bütün ədədlər a_i -lərin arasında sadəcə bir dəfə var. Qeyd edin ki, sakin köçmək istəməyə bilər, buna görə də $a_i = i$ -ə icazə verilir.

Sakinlər sadəcə bir dəfə evlərini dəyişə bilərlər. Bu o deməkdir ki, sabit test üçün N və a_i -lərin dəyəri proqramınızın bütün işləyişləri üçün eyni olacaq.

Birinci işləyiş.

Proqramınızın birinci işləyişi üçün $w = 0$ -dır. Bu işləyişdə, Durnanın evlərin yanından neçə dəfə keçməsinə istədiyinizi göstərən W ($0 \leq W \leq 60$) tam ədədini çapa verirsiniz. Proqramınız sonra sonlanmalıdır. Bundan sonra proqramınız W -dəfə yenidən çalışdırılacaqdır.

Sonrakı işləyişlər.

Proqramınız sonrakı işləyişində $w = 1$, ondan sonra $w = 2$, və beləliklə $w = W$ olana kimi davam işləyəcəkdir.

w , N və a_0, a_1, \dots, a_{N-1} -ləri oxuduqdan sonra, Durna küçədə yürüyüşünə başlayır.

- Əgər w çütdürsə, Durna evindən məktəbə gedir və evləri $0, 1, \dots, N - 1$ sırası ilə keçəcəkdir.

Bundan sonra proqramınız 0 -cı evin qarşısındakı heykəlin vəziyyətini göstərən b_0 ya 0 (bağlı) və ya 1 (açıq) olan sətiri oxumalıdır. b_0 -ı oxuduqdan sonra, b_0 -nin təyin etmək istədiyiniz yeni dəyəri olan 0 və ya 1 -i çap etməlisiniz.

Sonra proqramınız 1 -ci evin qarşısındakı heykəlin vəziyyətini göstərən b_1 -i oxumalıdır; və b_1 -nin yeni dəyərini çap etməlisiniz. Bu N -ci evə kimi davam edir. Durna sonuncu evi də ziyarət etdikdən sonra, **proqramınız sonlanmalıdır**.

Qeyd edin ki, proqramınız b_{i+1} -nin dəyərini sadəcə b_i -nin yeni dəyərini çap etdikdən sonra oxuya bilər.

- Əgər w tək ədədirsə, Durna məktəbdən evinə doğru gedir və o evləri tərs sırada $N - 1, N - 2, \dots, 0$ keçəcək. Beləliklə, siz birinci b_{N-1} -nin dəyərini oxuyub və yeni dəyərini çap edirsiniz, sonra b_{N-2} və belə b_0 -a kimi.

$w = 1$ olduğu zaman, giriş verilənləri b_0, b_1, \dots, b_{N-1} quş heykəllərinin orijinal vəziyyətləridir (həmçinin sakinlərin də üstünlük verdikləri vəziyyətdir). $w > 1$ olduğu zaman, proqramınıza daxil edilən b_0, b_1, \dots, b_{N-1} dəyərləri proqramınızın əvvəlki işində onları təyin etdiyi kimi olacaq.

Sonda, proqramınızın sonuncu işləyişindən sonra, bütün i -lər üçün b_i -nin dəyəri b_{a_i} -nin orijinal dəyəri olmalıdır, yoxsa "Səhv cavab"/"Wrong Answer" alacaqsınız.

Detallar.

Əgər proqramınızın $W + 1$ dəfə ayrı-ayrı işləyiş vaxtlarının cəmi verilən zaman limitini aşarsa "Zaman Limitini Aşma" / "Time Limit Exceeded" alacaqsınız.

Hər sətiri çap etdikdən sonra standart çıxışı sildiyyəinizə əmin olun, əks halda proqramınız "Zaman Limitini Aşma" kimi qiymətləndirilə bilər. Python-da `input()` istifadə etdiyiniz zaman avtomatik olaraq baş verir. C++-da `cout << endl;` yeni sətir çap etməkdən əlavə standart çıxışı silir. Əgər `printf` istifadə edirsinizsə, `fflush(stdout)` ilə buna nail ola bilərsiniz.

Məhdudiyyətlər və Qiymətləndirmə

- $2 \leq N \leq 500$.
- $W \leq 60$.

Həlliniz hər biri bir neçə bal dəyərində olan bir sıra test qrupları üzərində sınaqdan keçiriləcək. Hər bir test qrupu bir neçə testdən ibarətdir. Test qrupundan bal almaq üçün həmin test qrupundakı bütün testləri həll etməlisiniz.

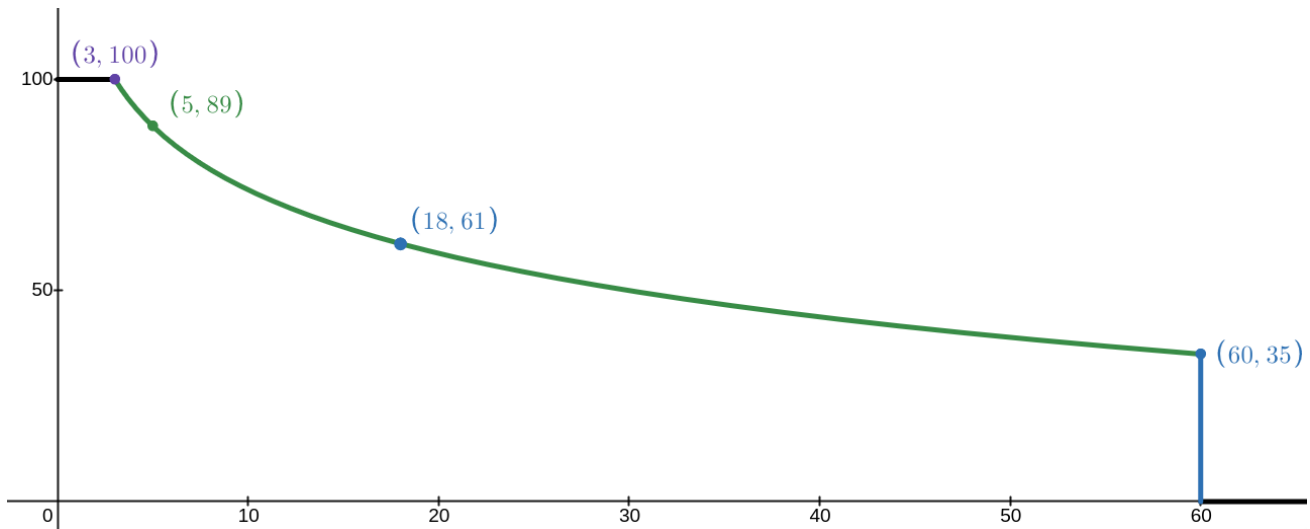
Qrup	Maksimum bal	Limitlər
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Əlavə məhdudiyyət yoxdur

Proqramınız hər bir test qrupu üçün doğru işləyirsə, aşağıda verilən düstura uyğun ballandırma olacaqdır:

$$\text{score} = S_g \cdot \left(1 - \frac{1}{2} \log_{10}(\max(W_g, 3)/3)\right),$$

hansı ki, S_g test qrupunun maksimum balıdır, və W_g , W -nin test qrupunda olan bütün testlər üçün istifadə olunan maksimum dəyəridir. Hər bir test qrupu üçün olan ballandırsa bir sonrakı tam ədədə yuvarlaşdırılacaqdır.

Aşağıdakı qrafik ballandırmanı göstərir. Xüsusilə, bu tapşırıq üzrə 100 bal toplamaq üçün hər bir testi $W \leq 3$ ilə həll etməlisiniz.



Yoxlama aləti

Həllinizi yoxlamağı asanlaşdırmaq üçün yükləyə biləcəyiniz sadə bir alət verilir. Kattis'də məsələnin olduğu səhifənin aşağısındakı "attachments" hissəsinə baxın. Aləti dəyişə və ya istifadə etməyə də bilərsiniz. Qeyd edək ki rəsmi qreyder sizə verilən yoxlama alətindən fərqli işləyir.

Aləti işlətmək üçün, boş bir fayl yaradın, "sample1.in", və aşağıdakı verilənləri fayla köçürün:

```
6
1 2 0 4 3 5
1 1 0 0 1 0
```

Python proqramları üçün, məsələnin faylın adı `solution.py` olsun (normalda `pypy3 solution.py` sətiri ilə işə salınır):

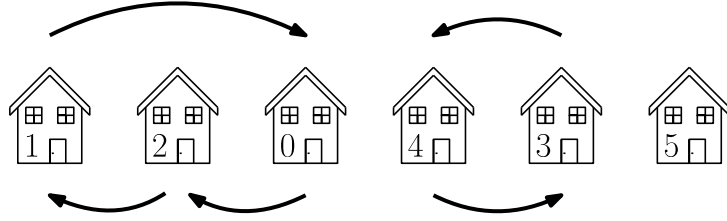
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ proqramları üçün, əvvəlcə kompilyat edin (məsələn `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) daha sonra işə salın:

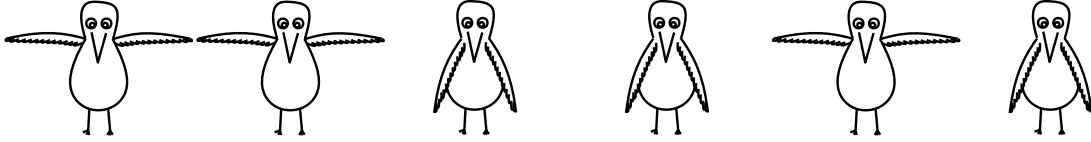
```
python3 testing_tool.py ./solution.out < sample1.in
```

Nümunə

Nümunədə, evlərdə qalan insanların permutasiyası verilmişdir:

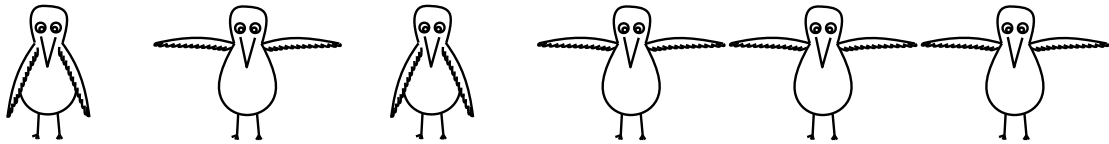


Proqram ilk dəfə işləyəndə ($w = 0$ olanda), o çapa $W = 2$ verir, bu da Durna küçədən iki dəfə keçəcək deməkdir (proqram da iki dəfə çalışdırılacaqdır). Küçədən ilk keçiddən qabaq, bağçalardalı quş heykəlləri aşağıdakı kimi görünür:



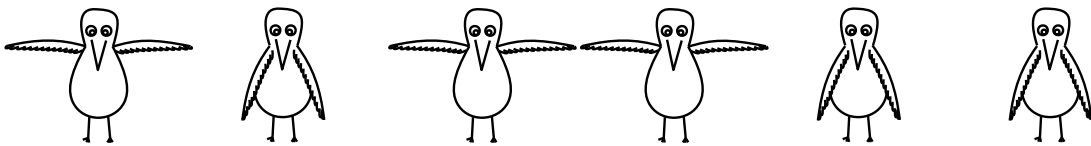
Sonrakı işləyişdə $w = 1$ olur, bu da Durnanın ilk küçədən keçidi olduğunu göstərir. O, soldan başlayaraq heykəllərin yanından tək tək keçir və ehtimalla onların vəziyyətini dəyişir. Nümunə proqram i -ci quş heykəlinin vəziyyətini çap etdikdən sonra $(i + 1)$ -ci quş heykəlinə keçir.

Durna məktəbə çatdıqdan sonra quş heykəlləri belə görünür:



Sonuncu işləyişdə ($w = 2$ olanda), Durna məktəbdən evə gəlir. Unutmayın ki, bu halda o, quşları sağdan sola keçəcək və onlara ardıcılıqla vaxt ayıracaq! Bu o deməkdir ki, o, $(i - 1)$ -ci quşu görməzdən əvvəl i -ci quşun vəziyyətini müəyyən etməlidir.

O, gəzintisini bitirdikdən sonra quş heykəlləri belə görünür:



Həqiqətən, bu düzgün konfigurasiyadır. Məsələn, quş heykəli 3 (yəni soldan dördüncü) açıqdır (indi $b_3 = 1$), bu düzgün haldır, çünki 4-cü sakin ora köçəcək ($a_3 = 4$) və onların əvvəlcə açıq quş heykəli var idi (əvvəlcə $b_4 = 1$).

qreyderin çıxışı	sizin çıxış
0 6	
1 2 0 4 3 5	
	2

qreyderin çixışı	sizin çixış
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

qreyderin çixışı	sizin çixış
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1