

D. Garden Decorations

Numele problemei	Garden Decorations
Limita de timp	7 seconds
Limita de memorie	1 gigabyte

În fiecare zi, când merge la școală și se întoarce acasă, Detje parcurge o stradă cu case N , numerotate de la 0 la $N - 1$. În prezent, casa i este locuită de persoana i . Pentru o schimbare de peisaj, locuitorii au decis să-și schimbe locuința între ei. Persoana care se va muta în casa i este persoana a_i (care locuiește în prezent în casa a_i).

Fiecare casă are în grădină o statuie reprezentând o pasăre. Statuile au două stări posibile: fie au aripile *deschise* (ca și cum pasărea zboară), fie *închise* (ca și cum ar sta pe pământ). Locuitorii au preferințe foarte puternice cu privire la modul în care ar trebui să arate statuile lor de păsări. Momentan pasărea din fața casei i este în starea preferată de rezidentul i . Rezidenții refuză să se mute într-o casă până când statuia nu e aranjată așa cum preferă. Detje vrea să-i ajute să aranjeze statuile păsărilor astfel încât să se poată muta.

Pentru aceasta, ea face următoarele: ori de câte ori parcurge strada (fie în drum spre școală, fie înapoi acasă), ea observă păsările pe lângă care trece una câte una și, eventual, ajustează unele dintre statui (prin deschiderea sau închiderea aripilor). Întrucât zilele ei la școală și acasă sunt foarte aglomerate, **nu își amintește stările păsărilor pe care le-a văzut în parcurgerile anterioare**. Din fericire, ea a notat lista a_0, a_1, \dots, a_{N-1} , așa că știe fiecare rezident unde se mută.

Ajut-o pe Detje să creeze o strategie care îi spune ce păsări să manipuleze pentru a ajusta statuile după preferințele rezidenților. Poate parcurge strada de cel mult 60 ori, dar pentru a obține un scor mai mare, ar trebui să parcurgă stradă de mai puține ori.

Implementare

Aceasta este o problemă cu mai multe rulări, ceea ce înseamnă că programul dumneavoastră va fi executat de mai multe ori.

La fiecare rulare, ar trebui să citiți mai întâi o linie cu două numere întregi w și N , indicele parcurgerii și numărul de case.

La prima rulare a programului dvs. $w = 0$, în a doua $w = 1$ și așa mai departe (mai multe detalii sunt explicate mai jos).

Pe a doua linie a inputului sunt N numere întregi a_0, a_1, \dots, a_{N-1} , ceea ce înseamnă că persoana care se va muta în casa i locuiește în prezent în casa a_i . Șirul a_i -urilor formează o *permutare*: adică fiecare număr de la 0 la $N - 1$ apare exact o dată în șirul a_i -urilor. Rețineți că, un rezident poate alege să nu se mute; adică $a_i = i$ este permis.

Locuitorii își schimbă locuința o singură dată. Aceasta înseamnă că pentru un test fixat, valoarea N și șirul a_i -urilor vor fi aceleași pentru toate rulările programului dumneavoastră.

Prima rulare.

La prima execuție a programului dvs., $w = 0$. La această rulare, ar trebui să afișați un singur număr întreg W ($0 \leq W \leq 60$), de câte ori doriți ca Detje să parcurgă strada pe lângă case. Programul dvs. ar trebui apoi să se oprească. După aceasta, programul dumneavoastră va fi executat din nou de W ori.

Rulări ulterioare.

La următoarea rulare a programului dvs., $w = 1$; în următoarea $w = 2$; și așa mai departe până la rularea finală unde $w = W$.

După ce ați citit w , N și a_0, a_1, \dots, a_{N-1} , Detje începe să parcurgă strada.

- Dacă w este impar, Detje merge de acasă la școală și va trece pe lângă case în ordinea $0, 1, \dots, N - 1$.

Programul dvs. ar trebui să citească acum o linie cu b_0 , fie 0 (închis), fie 1 (deschis), starea actuală a statuii din fața casei 0. După ce citiți b_0 , ar trebui să afișați o linie cu 0 sau 1, reprezentând noua valoare la care doriți să setați b_0 .

Apoi, programul dvs. ar trebui să citească o linie cu b_1 , starea statuii din fața casei 1; și afișați noua valoare pentru b_1 . Aceasta continuă pentru fiecare dintre cele N case. După ce Detje trece de casa finală (adică, citești și scrii b_{N-1}) programul tău ar trebui să se oprească. *Rețineți că programul dvs. poate citi următoarea valoare b_{i+1} numai după ce ați scris noua valoare a lui b_i .*

- Dacă w este par, Detje merge de la școală la casa ei și va trece în schimb pe lângă case în ordine inversă $N - 1, N - 2, \dots, 0$.

De aceea, începeți cu citirea și scrierea b_{N-1} , apoi b_{N-2} și așa mai departe până la b_0 .

Când $w = 1$, valorile de intrare b_0, b_1, \dots, b_{N-1} sunt starea inițială a statuilor păsărilor (acestea sunt și stările preferate de rezidenți). Când $w > 1$, valorile de intrare b_0, b_1, \dots, b_{N-1} vor fi cele pe care le-a stabilit rularea anterioară a programului dumneavoastră.

La sfârșit, după rularea finală a programului dvs., valoarea lui b_i trebuie să fie egală cu valoarea inițială a b_{a_i} pentru toți i , altfel veți primi verdictul Răspuns greșit.

Detalii.

Dacă *suma* duratelor de rulare $W + 1$ ale programului dvs. depășește limita de timp, submisia dvs. va fi considerată având Limită de timp depășită.

Asigurați-vă că faceți flush după tipărirea fiecărei linii, în caz contrar programul dvs. ar putea fi considerat ca limită de timp depășită. În Python, acest lucru se întâmplă automat atâta timp cât utilizați `input()` să citiți linii. În C++, `cout << endl;` dă flush, pe lângă afișatul unei linii noi. Dacă folosiți `printf`, utilizați `fflush(stdout)`.

Restricții și Punctaj

- $2 \leq N \leq 500$.
- Poți utiliza cel mult $W \leq 60$ runde de rulări.

Soluția va fi testată pe un set de subtaskuri, fiecare valorând un număr de puncte. Fiecare subtask conține un set de teste. Pentru a obține punctele pentru un subtask, trebuie rezolvate toate testele conținute.

Subtask	Punctaj maxim	Restricții
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	Fără restricții adiționale

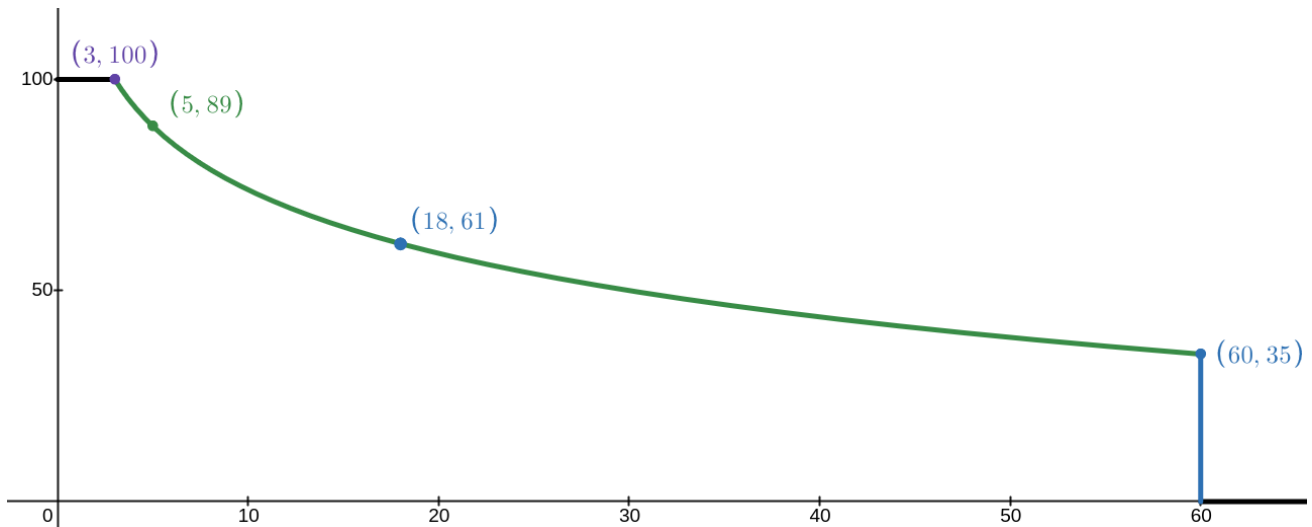
Pentru fiecare subtask pe care programul dumneavoastră îl rezolvă corect, veți primi un scor pe baza următoarei formule

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

unde S_g este scorul maxim pentru subtask și W_g este valoarea maximă a lui W utilizat pentru orice test din subtask. Scorul tău pentru fiecare subtask va fi rotunjit la cel mai apropiat număr întreg.

Graficul de mai jos arată numărul de puncte, în funcție de W , pe care programul dumneavoastră îl va obține dacă rezolvă toate subtaskurile cu aceeași valoare a lui W . În special, pentru a obține un

scor de 100 de puncte la această problemă, trebuie să rezolvați fiecare caz de testare cu $W \leq 3$.



Testing Tool

Pentru a facilita testarea soluției dvs., vă oferim un instrument simplu pe care îl puteți descărca. Consultați „atașamente” în partea de jos a pagina Kattis a problemei. Instrumentul este opțional de utilizat. Rețineți că grader-ul oficial pe Kattis este diferit de instrumentul de testare.

Pentru a utiliza instrumentul, creați un fișier de intrare, cum ar fi „sample1.in”, care ar trebui să înceapă cu un număr N urmat de o linie cu numere N care specifică permutarea și o altă linie cu N biți (0 sau 1) specificând stările inițiale ale păsărilor. De exemplu:

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

Pentru programele Python, `solution.py` (`python3 solution.py`):

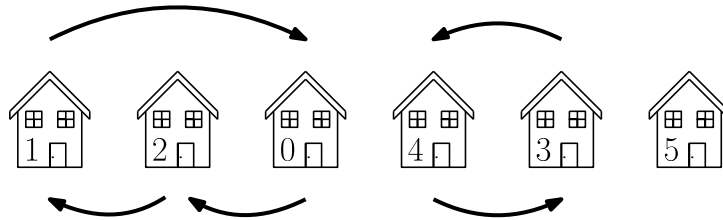
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Pentru programele C++, mai întâi compilați (e.g. with `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) și apoi rulați:

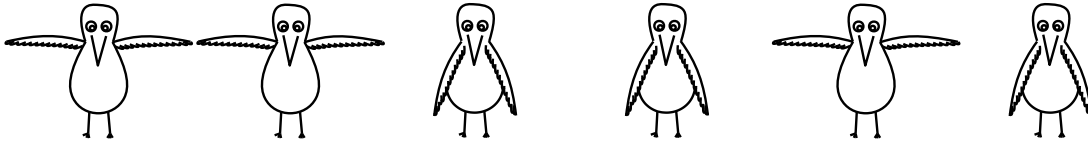
```
python3 testing_tool.py ./solution.out < sample1.in
```

Example

În exemplu, ni se dă următoarea permutare a oamenilor din case

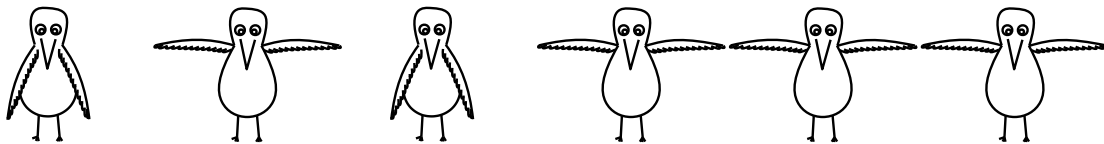


Prima dată când se rulează programul (cu $w = 0$), acesta afișează $W = 2$, ceea ce înseamnă că Detje va merge pe stradă de două ori (și programul va fi rulat încă de două ori). Înainte de prima parcurgere, păsările din grădini arată astfel:



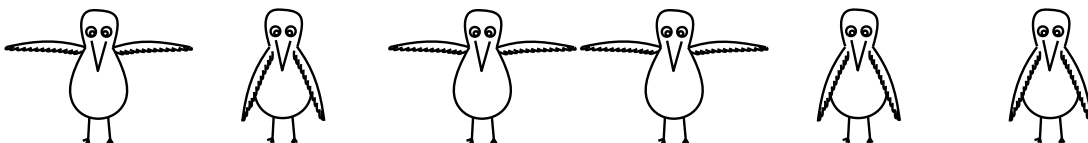
Apoi programul este rulat cu $w = 1$: indicând prima parcurgere a lui Detje. Ea trece de-a lungul păsărilor, una câte una, începând din stânga și, eventual, îi schimbă starea. Programul trebuie să afișeze starea păsării i înainte de a vedea $(i + 1)$ -a pasăre.

După ce Detje a ajuns la școală, stările păsărilor sunt:



La rularea finală a programului (cu $w = 2$), Detje se întoarce acasă de la școală. Amintiți-vă că, în acest caz, ea va trece prin dreptul păsărilor de la dreapta la stânga și le va procesa în această ordine! Aceasta înseamnă că trebuie să determine starea i -a pasăre înainte de a vedea $(i - 1)$ -a pasăre.

După ce termină parcurgerea, păsările arată acum astfel:



Într-adevăr, aceasta este configurația corectă. De exemplu, statuia păsării 3 (adică a patra din stânga) este deschisă (acum $b_3 = 1$), ceea ce este corect deoarece persoana 4 se va muta acolo ($a_3 = 4$) și ea avea inițial o statuie cu pasărea cu aripi deschise (inițial $b_4 = 1$).

grader output	your output
0 6	
1 2 0 4 3 5	
	2

grader output	your output
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

grader output	your output
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1