D. Vrtni ukrasi

Ime zadatka	Garden decorations
Vremensko ograničenje	7 sekundi
Memorijsko ograničenje	1 gigabajt

Svaki dan kada ide u školu i kući, Lara šeta ulicom s N kuća, označenih brojevima od 0 do $N-1$. Trenutno u kući i živi osoba i. Za promjenu okruženja, stanovnici su odlučili međusobno zamijeniti kuće. Osoba koja će se useliti u kuću i je osoba a_i (koja trenutno živi u kući a_i).

Svaka kuća ima statuu ptice identičnog izgleda u vrtu. Kipovi imaju dva moguća stanja: *otvorena* krila (kao da ptica leti) ili *zatvorena* (kao da stoji na zemlji). Stanovnici imaju vrlo jake preferencije o tome kako bi njihove statue ptica trebale izgledati, i odbijaju se preseliti u svoju novu kuću prije nego što kip u njihovom novom vrtu izgleda kao u njihovom prethodnom vrtu. Lara, gledajući na ovu situaciju kao fora zagonetku, želi im pomoći urediti kipove ptica kako bi se mogli preseliti.

Da bi to učinila, radi sljedeće: kad god šeta ulicom (bilo na putu do škole ili kući), promatra ptice kraj kojih prolazi i eventualno namješta neke od kipova (otvaranjem ili zatvaranjem krila). Budući da u školi i kod kuće često razmišlja o tome koja riječ ima isti korijen u najviše jezika (druga zagonetka koju joj je zadala prijateljica Lana), ne sjeća se više stanja ptica koje je vidjela u prijašnjim šetnjama. Srećom, zapisala je popis $a_0, a_1, \ldots, a_{N-1}$ pa zna koji se stanar kamo seli.

Pomozite Lari da osmisli strategiju koja joj govori kojim pticama treba manipulirati kako bi prilagodila kipove željama stanovnika. Ona može hodati ulicom najviše 60 puta, ali da bi postigla veći rezultat, trebala bi hodati ulicom manje puta.

Implementacija

Ovo je *multirun* zadatak, što znači da će se vaš program izvršavati više puta.

Pri svakom pokretanju prvo treba učitati redak s dva cijela broja w i N, indeks hoda i broj kuća. U prvom pokretanju vašeg programa $w = 0$, u drugom $w = 1$, i tako dalje (detaljnije objašnjeno dolje).

U drugom retku unosa nalazi se N cijelih brojeva $a_0, a_1, \ldots, a_{N-1}$, što znači da osoba koja će se useliti u kuću i trenutno živi u kući a_i. a_i-evi čine *permutaciju*: to jest, svaki broj od 0 do $N-1$

pojavljuje se točno jednom na popisu a_i-eva. Imajte na umu da stanovnik može odlučiti da se ne preseli; odnosno dozvoljeno je $a_i = i$.

Stanovnici samo jednom mijenjaju kuće. To znači da će za fiksni testni slučaj vrijednost N i popis a_is biti isti za sva izvođenja vašeg programa.

Prvo izvođenje (first run)

Za prvo izvođenje vašeg programa, $w = 0$. U ovom izvođenju, trebali biste jednostavno ispisati jedan cijeli broj W ($0 \leq W \leq 60$), koliko puta želite da Lara prođe pored kuća. Vaš bi program tada trebao završiti. Nakon toga, vaš program će se ponovno izvršiti još W puta.

Naknadna izvođenja (subsequent runs)

U sljedećem izvođenju vašeg programa će biti $w = 1$; u izvođenju nakon toga $w = 2$; i tako dalje do finalnog izvođenja gdje je $w = W$.

Nakon što ste učitali w, N i $a_0, a_1, ..., a_{N-1}$, Lara počinje hodati ulicom.

- Ako je w neparan, Lara hoda od svoje kuće do škole i proći će pokraj kuća redoslijedom $0, 1, ..., N-1$.

  Vaš program bi sada trebao učitati redak s b_0, koji je ili 0 (zatvoreno) ili 1 (otvoreno) i označava trenutno stanje kipa ispred kuće 0. Nakon što učitate b_0, trebali biste ispisati redak s 0 ili 1, novom vrijednošću na koju želite postaviti b_0.

  Zatim bi vaš program trebao učitati red s b_1, stanjem kipa ispred kuće 1; i ispisati novu vrijednost b_1. Ovo se nastavlja za svaku od N kuća. Nakon što prođete posljednju kuću (tj. učitate i ispišete b_{N-1}) vaš program bi trebao završiti.

  Imajte na umu da vaš program može jedino učitati sljedeću vrijednost b_{i+1} nakon što ste ispisali vrijednost b_i.

- Ako je w paran, Lara se vraća iz škole prema svojoj kući te će onda proći pokraj kuća obrnutim redoslijedom $N-1, N-2, ...0$.

  Proces je isti kao kada je w neparan, osim što počinjete s čitanjem i ispisivanjem b_{N-1}, zatim b_{N-2}, i tako dalje sve do b_0.

Kada je $w = 1$, ulazne vrijednosti $b_0, b_1, ..., b_{N-1}$ su izvorna stanja kipova ptica. Kada je $w > 1$, ulazne vrijednosti $b_0, b_1, ..., b_{N-1}$ u vaš program bit će ono na što ih je prethodno izvođenje vašeg programa postavilo.

Na kraju, nakon konačnog izvođenja vašeg programa, vrijednost b_i mora biti jednaka izvornoj vrijednosti b_{a_i} za sve i, inače ćete dobiti presudu Wrong Answer.

Detalji.

Ako *zbroj* vremena izvođenja $W+1$ zasebnih pokretanja vašeg programa premaši vremensko ograničenje, vaša prijava bit će ocijenjena kao Time Limit Exceeded.

Provjerite jeste li ispraznili standardni izlaz nakon ispisa svakog retka, inače bi vaš program mogao biti ocijenjen kao Time Limit Exceeded. U Pythonu se to događa automatski sve dok koristite `input()` za čitanje redaka. U C++, `cout << endl;` isto to radi automatski uz ispis novog retka; ako koristite printf, koristite `fflush(stdout)`.

Ograničenja i bodovanje

- $2 \leq N \leq 500$.
- Možete koristiti najviše $W \leq 60$ rundi.

Vaše rješenje bit će testirano na nizu testnih grupa, gdje svaka nosi nekoliko bodova. Svaka testna grupa sastoji se od niza testnih primjera. Da bi dobili bodove za testnu grupu trebate riješiti sve testne primjere koji pripadaju toj testnoj grupi.

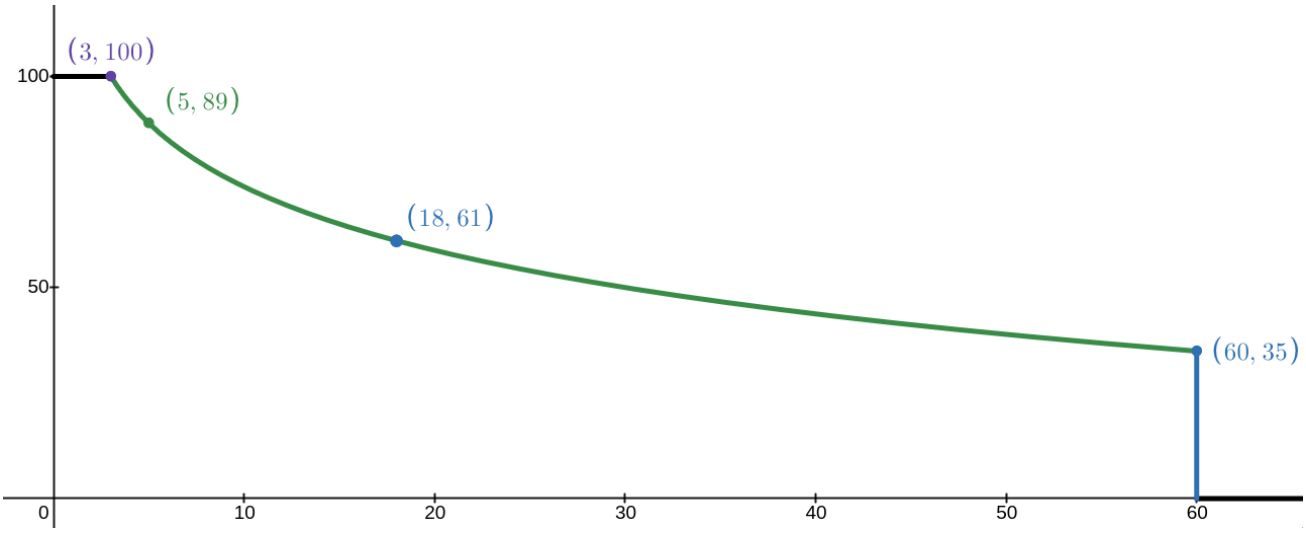
Grupa	Max bodova	Ograničenja
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Nema dodatnih ograničenja

Za svaku testnu grupu koju vaš program točno riješi dobit će te bodove prema sljedećoj formuli:

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

gdje je S_g S_g maksimalni rezultat za testnu grupu, a W_g maksimalna vrijednost od W kroz sve test primjere u toj test grupi. Vaš rezultat za svaku testnu grupu bit će zaokružen na najbliži cijeli broj.

Dijagram u nastavku prikazuje broj bodova, kao funkciju W, koje će vaš program dobiti ako riješi sve test grupe s istom vrijednošću W. Konkretno, da biste postigli rezultat od 100 bodova na ovom problemu, trebate riješiti svaki testni slučaj s $W \leq 3$.



Alat za testiranje (testing tool)

Kako bismo olakšali testiranje vašeg rješenja, nudimo jednostavan alat koji možete preuzeti. Pogledajte “attachments” na dnu stranice Kattis problema. Alat je neobavezan za korištenje. Imajte na umu da se službeni program ocjenjivača na Kattisu razlikuje od alata za testiranje.

Da biste koristili alat, izradite ulaznu datoteku, kao što je "sample1.in", koja bi trebala započeti s brojem N nakon čega slijedi redak s N brojeva koji određuju permutaciju i drugi redak s N bitova (0 ili 1) specificirajući početna stanja ptica. Na primjer:

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

Za Python programe, recimo `solution.py` (obično pokrećemo sa `pypy3 solution.py`):

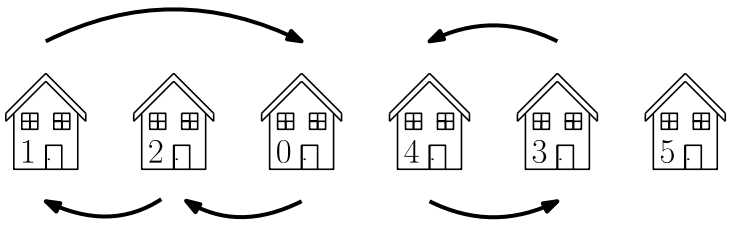
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Za C++ programe, prvo kompajlirajte (e.g. with `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) i onda pokrenite:

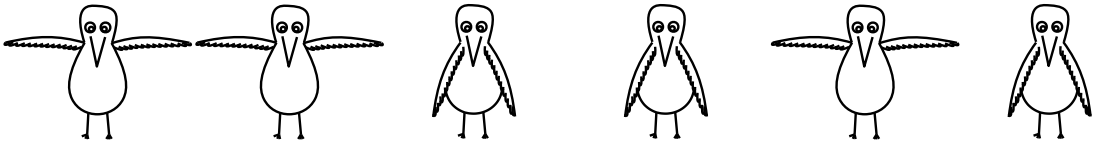
```
python3 testing_tool.py ./solution.out < sample1.in
```

Primjer

U probnom primjeru nam je zadana sljedeća permutacija ljudi u kućama:

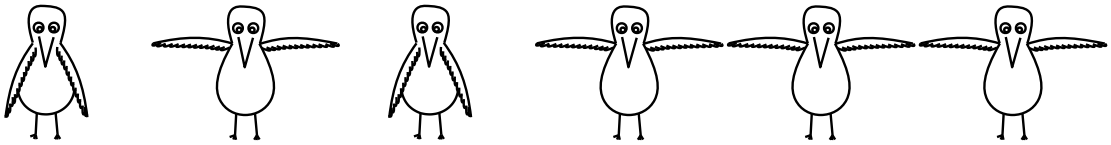


Prvi put kada se ogledni program pokrene (s $w = 0$), on ispisuje $W = 2$, što znači da će Lara prošetati ulicom dva puta (i program će se pokrenuti još dva puta). Prije prve šetnje, ptice u vrtovima izgledaju ovako:



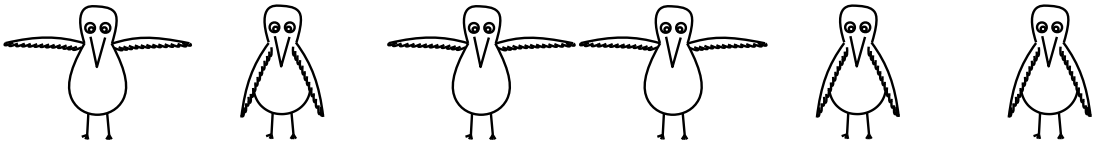
Zatim se program izvodi s $w = 1$: označavajući Larinu prvu šetnju. Ona prolazi kroz ptice jednu po jednu, počevši s lijeve strane, i eventualno mijenja njihovo stanje. Uzorak programa mora ispisati stanje i-te ptice prije nego što vidimo $(i + 1)$-tu pticu.

Nakon što je Lara stigla u školu, stanje ptica izgleda ovako:



U konačnom izvođenju programa (s $w = 2$), Lara se vraća kući iz škole. Upamtite da će u ovom slučaju ona proći kroz ptice zdesna nalijevo i obraditi ih obrnutim redoslijedom! To znači da mora odrediti stanje i-te ptice prije nego što vidi $(i - 1)$-tu pticu.

Nakon što završi svoju šetnju, ptice sada izgledaju ovako:



Doista, ovo je ispravna konfiguracija. Na primjer, kip ptice 3 (tj. četvrte slijeva) je otvoren (sada je $b_3 = 1$), što je točno jer će se osoba 4 tamo preseliti ($a_3 = 4$) i izvorno je imala otvoreni kip ptice (izvorno $b_4 = 1$).

izlaz ocjenjivača	vaš izlaz
0 6	
1 2 0 4 3 5	
	2

izlaz ocjenjivača	vaš izlaz
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

izlaz ocjenjivača	vaš izlaz
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1