

D. Vrtne Dekoracije

Naziv Problema	Vrtne Dekoracije
Vremensko Ograničenje	7 sekundi
Memorijsko Ograničenje	1 gigabajt

Svaki dan kada ide u školu i vraća se kući, Detje prolazi ulicom sa N kuća, označenih brojevima od 0 do $N - 1$. Trenutno, u kući i stanuje osoba i . Za promjenu ambijenta, stanovnici su odlučili da zamijene kuće jedni s drugima. Osoba koja će se useliti u kuću i je osoba a_i (koja trenutno živi u kući a_i).

Svaka kuća ima kip ptice u bašti. Kipovi imaju dva moguća stanja: krila su im *otvorena* (kao da ptica leti) ili *zatvorena* (kao da stoji na zemlji). Stanovnici imaju vrlo snažne preferencije o tome kako njihovi ptičji kipovi trebaju izgledati, i odbijaju se useliti u novu kuću dok kip u njihovoj novoj bašti ne izgleda kao što je izgledao u njihovoj prethodnoj bašti. Detje želi da im pomogne da uredi kipove kako bi se mogli preseliti.

Kako bi ovo postigla, radi sljedeće: kad god prolazi ulicom (bilo na putu do škole ili nazad kući), jedan po jedan posmatra kipove ptica kraj kojih prolazi i eventualno podešava neke od kipova (otvarajući ili zatvarajući njihova krila). Pošto su njeni dani u školi i kod kuće vrlo zauzeti, ne sjeća se stanja ptica koje je vidjela na prethodnim šetnjama. Srećom, zapisala je niz a_0, a_1, \dots, a_{N-1} , tako da zna koja osoba se seli gdje.

Pomozite Detje da osmisli strategiju koja će joj reći koje ptice da izmijeni kako bi podesila kipove po željama stanovnika. Detje može proći ulicom najviše 60 puta, ali da bi dostigla veći broj bodova, treba proći ulicom što manje puta.

Implementacija

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

U svakom izvođenju, prvo trebate učitati liniju sa dva cijela broja w i N , indeksom šetnje i brojem kuća. U prvom izvođenju vašeg programa biti će $w = 0$, u drugom će biti $w = 1$, i tako dalje (detaljnije pojašnjeno u nastavku teksta).

Na drugoj liniji ulaza nalazi se N cijelih brojeva a_0, a_1, \dots, a_{N-1} . Ovo označava da osoba koja će se useliti u kuću i trenutno živi u kući a_i . Primijetite da a_i -evi čine *permutaciju*: to jest, svaki broj od 0

do $N - 1$ pojavljuje se tačno jednom u nizu a_i -eva. Napomena: stanar može odlučiti da se ne seli; to jest, dozvoljeno je da $a_i = i$.

Stanari se sele samo jednom. To znači da za fiksni testni slučaj, vrijednost N i niz a_i -eva će biti iste za sva izvođenja vašeg programa.

Prvo Izvođenje.

U prvom izvođenju vašeg programa, $w = 0$. U ovom izvođenju, trebate jednostavno ispisati cijeli broj W ($0 \leq W \leq 60$), broj puta koliko želite da Detje prođe pored kuća. Vaš program tada treba da se završi. Nakon toga, vaš program će se izvoditi još W puta.

Naredna Izvođenja.

U narednom izvođenju vašeg programa biti će $w = 1$; u onom nakon toga bit će $w = 2$; i tako dalje do konačnog izvođenja gdje će biti $w = W$.

Nakon što unesete w , N i a_0, a_1, \dots, a_{N-1} , Detje počinje šetnju ulicom.

- Ako je w neparan, Detje ide od svoje kuće do škole i prolazi pored kuća redosljedom $0, 1, \dots, N - 1$. Vaš program sada treba da učita liniju sa b_0 koji označava trenutno stanje kipa ispred kuće 0. Ako je b_0 jednak 0, ptičiji kip ima zatvorena krila, a ako je 1 ptičiji kip ima otvorena krila. Nakon što učitate b_0 , trebate ispisati liniju sa ili 0 ili 1, novom vrijednošću na koju želite postaviti b_0 . Zatim vaš program treba da učita liniju sa b_1 , stanjem kipa ispred kuće 1; i ispiše novu vrijednost za b_1 . Ovo se nastavlja za svaku od N kuća. Nakon što prođete zadnju kuću (tj. učitate i ispišete b_{N-1}) vaš program treba da se završi. *Napomena da vaš program može pročitati narednu vrijednost b_{i+1} tek nakon što ste ispisali vrijednost b_i .*
- Ako je w paran, Detje ide od škole do svoje kuće, i prolazi pored kuća obrnutim redosljedom $N - 1, N - 2, \dots, 0$. Proces je isti kao kada je w neparan, osim što počinjete sa čitanjem i ispisivanjem b_{N-1} , zatim b_{N-2} , i tako dalje dok ne dođete do b_0 .

Kada je $w = 1$, ulazne vrijednosti b_0, b_1, \dots, b_{N-1} su originalno stanje ptičjih kipova. Kada je $w > 1$, ulazne vrijednosti b_0, b_1, \dots, b_{N-1} će biti one koje je prethodno izvođenje vašeg programa postavilo.

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

Detalji.

Ako *zbir* količina vremena koje su trebale da bi se izvelo $W + 1$ odvojenih izvođenja vašeg programa premaši vremensko ograničenje, vaš submission će biti ocijenjen sa Time Limit Exceeded.

Pobrinite se da flush-ate standardni izlaz nakon ispisivanja svake linije, inače vaš program može biti ocijenjen sa Time Limit Exceeded. U Pythonu, ovo se dešava automatski dok god koristite `input()` za čitanje linija. U C++, `cout << endl;` osim što ispisuje novi red također flush-a; a ako koristite `printf`, koristite `fflush(stdout)`.

Ograničenja i Bodovanje

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

Vaše rješenje će biti testirano na skupu testnih grupa, svaka od kojih vrijedi određen broj bodova. Svaka testna grupa sadrži set testnih slučajeva. Da biste dobili bodove za testnu grupu, morate riješiti sve testne slučajeve u toj grupi.

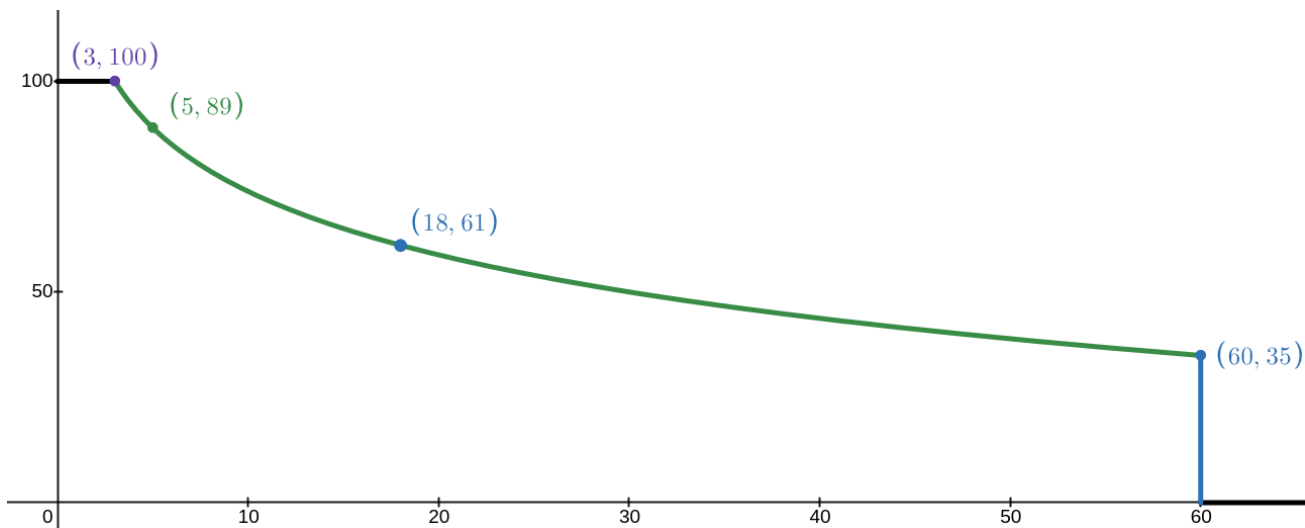
Grupa	Maksimalan broj bodova	Ograničenja
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Bez dodatnih ograničenja

Za svaku testnu grupu koju vaš program ispravno riješi, dobit ćete bodove na osnovu sljedeće formule:

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

gdje je S_g maksimalan broj bodova za testnu grupu, a W_g je maksimalna vrijednost W -a korišten u bilo kojem testnom primjeru u testnoj grupi. Vaš ukupan broj bodova za svaku test grupu će biti zaokružen na najbliži cijeli broj.

Grafikon ispod prikazuje broj bodova, u funkciji od W , koji će vaš program dobiti ako bi riješio sve testne grupe sa istom vrijednošću W . Konkretno, da biste postigli rezultat od 100 poena na ovom zadatku, morate svaki testni slučaj riješiti sa $W \leq 3$.



Alat za Testiranje

Da bi vam olakšali testiranje rješenja, pružamo jednostavan alat koji možete preuzeti. Pogledajte "attachments" na dnu Kattis stranice za zadatak. Korištenje alata nije obavezno. Imajte na umu da je zvanični program za ocjenjivanje na Kattisu drugačiji od alata za testiranje.

Da biste koristili alat, kreirajte ulaznu datoteku, npr. "sample1.in", koja treba počinjati s brojem N . nakon čega slijedi linija s N brojeva koji specificiraju permutaciju, te još jedna linija s N bitova (0 ili 1) koji specificiraju 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 se pokreće sa `pypy3 solution.py`):

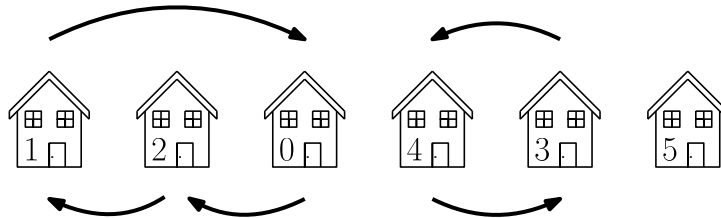
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Za C++ programe, prvo ih kompajlirajte (npr. sa `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) a zatim pokrenite:

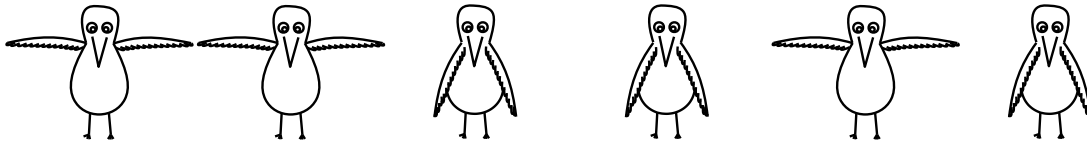
```
python3 testing_tool.py ./solution.out < sample1.in
```

Primjer

U primjeru, data nam je sljedeća permutacija osoba u kućama:

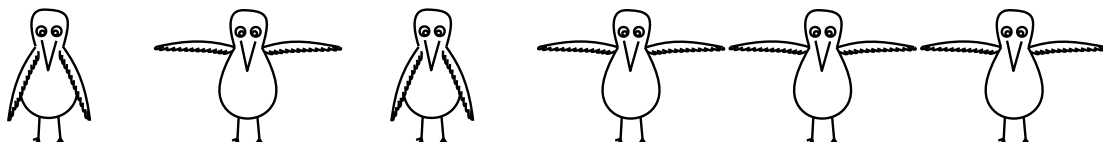


Prvi put kada se program za primjer pokrene (sa $w = 0$), on ispisuje $W = 2$, što znači da će Detje hodati duž ulice dva puta (i program će se pokrenuti još dva puta). Prije prve šetnje, ptice u vrtovima izgledaju ovako:



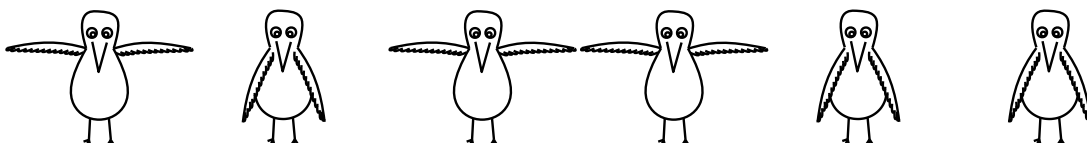
Zatim se program pokreće sa $w = 1$: što označava prvu šetnju Detje. Ona prolazi pored ptica jednu po jednu, počevši s lijeve strane, i po potrebi mijenja njihovo stanje. Primjerni program mora ispisati stanje i -te ptice prije nego što vidimo $(i + 1)$ -tu pticu.

Nakon što Detje stigne u školu, stanje ptica izgleda ovako:



U završnom pokretanju programa (sa $w = 2$), Detje se vraća kući iz škole. Zapanite da u ovom slučaju ona prolazi pored ptica s desna na lijevo i procesuiru 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:



I zaista, ovo je tačna konfiguracija. Na primjer, ptičiji kip broj 3 (tj. četvrti s lijeve strane) je otvoren (sada $b_3 = 1$), što je ispravno jer će se osoba 4 preseliti u kuću broj 3 ($a_3 = 4$), a ona je prvobitno imala otvoren pričiji kip (prvobitno $b_4 = 1$).

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