

D. Dārza dekorācijas

Uzdevuma ID	Garden decorations
Laika limits	7 sekundes
Atmiņas limits	1 gigabaits

Katru dienu, ejot uz skolu un atpakaļ mājās, Detje iet pa ielu, uz kuras ir N mājas, kas sanumurētas no 0 līdz $N - 1$. Šobrīd i -tajā mājā dzīvo i -tais iemītnieks. Lai pamainītu ainavu, iemītnieki ir izlēmuši cits ar citu apmainīties mājām. i -tajā mājā ievāksies a_i -tais iemītnieks (kurš šobrīd dzīvo a_i -tajā mājā).

Katrai mājai dārzā ir putna statuja. Statujām ir iespējami divi stāvokļi - to spārni ir vai nu *izplesti* (it kā putns lidotu) vai *sakļauti* (it kā tas stāvētu uz zemes). Iemītniekiem ir ļoti stingras prasības tam, kā viņu putnu statujām vajadzētu izskatīties. Šobrīd putns, kas atrodas pie i -tās mājas, ir i -tā iemītnieka iecienītajā stāvoklī. Iemītnieki ir ar mieru ievākties mājā tikai tad, ja tās statuja atrodas viņu iecienītajā stāvoklī. Detje vēlas viņiem palīdzēt sakārtot putnu statujas, lai viņi varētu pārvākties.

Lai to panāktu, viņa rīkojas šādi: katru reizi, kad viņa iet pa ielu (vai nu uz skolu, vai atpakaļ mājup), viņa vēro putnus, kuriem viņa iet garām, un, iespējams, izmaina stāvokli daļai statuju (izplešot vai sakļaujot to spārnus). Tā kā gan skolā, gan mājās viņa ir ļoti aizņemta, **viņa neatceras stāvokli putniem, ko viņa redzēja iepriekšējās pastaigās**. Par laimi, viņa ir pierakstījusi sarakstu a_0, a_1, \dots, a_{N-1} , tāpēc viņa zina, kurš iemītnieks uz kurieni pārvāksies.

Palīdzi Detjei izdomāt stratēģiju, kuriem putniem izmainīt stāvokli, lai statujas atbilstu iemītnieku vēlmēm. Viņa var iet pa ielu ne vairāk kā 60 reizes, bet, lai iegūtu vairāk punktu, reižu skaitam jābūt mazākam.

Implementācija

Tava programma tiks izpildīta vairākas reizes.

Katrā izpildes reizē Tev vispirms jāielasa viena rinda, kurā ir divi veseli skaitļi w un N - gājiena numurs un māju skaits. Pirmajā programmas izpildes laikā $w = 0$, otrajā $w = 1$ utt. (tālāk tiek paskaidrots sīkāk).

Otrajā ievaddatu rindā ir N veseli skaitļi a_0, a_1, \dots, a_{N-1} , kas norāda, ka cilvēks, kurš ievāksies i -tajā mājā, šobrīd dzīvo a_i -tajā mājā. Skaitļi a_i veido *permutāciju*: tas ir, katrs skaitlis no 0 līdz $N - 1$ skaitļu a_i sarakstā parādās tieši vienreiz. Ņem vērā, ka iemītnieks var izlemt nepārvākties, tas ir, $a_i = i$ ir atļauts.

Iemītnieki maina mājas tikai vienreiz. Tas nozīmē, ka konkrētam testam N vērtība un skaitļu a_i saraksts visām programmas izpildes reizēm būs vienāds.

Pirmā izpildes reize.

Pirmajai programmas izpildes reizei $w = 0$. Šajā izpildes reizē, Tev tikai jāizvada viens vesels skaitlis W ($0 \leq W \leq 60$) - skaits, cik reižu Tu vēlētos, lai Detje iet gar mājām. Tad programmai jābeidz darbs. Pēc tam programma tiks atkārtoti izpildīta vēl W reizes.

Nākamās izpildes reizes.

Nākamajā programmas izpildes reizē $w = 1$, aiznākamajā $w = 2$ utt. līdz pēdējai izpildes reizei, kur $w = W$.

Pēc tam, kad esi ielasījis w , N un sarakstu a_0, a_1, \dots, a_{N-1} , Detje sāk iet pa ielu.

- Ja w ir nepāra, Detje iet no savām mājām uz skolu un viņa ies garām mājām šādā secībā: $0, 1, \dots, N - 1$.

Tad Tavai programmai jāielasa rinda, kurā ir skaitlis b_0 - vai nu 0 (sakļauti), vai 1 (izplesti) - statujas, kas atrodas 0. mājas priekšā, stāvoklis. Pēc tam, kad Tu ielasi b_0 , Tev jāizvada rinda ar vai nu 0, vai 1 - jaunais stāvoklis, kādā Tu vēlies nostādīt statuju b_0 .

Tad Tavai programmai jāielasa rinda ar b_1 - stāvokli, kādā ir statuja, kas atrodas 1. mājas priekšā; un jāizvada jaunā b_1 vērtība. Līdzīgi notiek katrai no N mājām. Pēc tam, kad Detje noiet garām pēdējai mājai (t.i., Tu ielasi b_{N-1}), **Tavai programmai jābeidz darbs.**

Ņem vērā, ka Tava programma var ielasīt nākamo vērtību b_{i+1} tikai pēc tam, kad ir izvadīta jaunā b_i vērtība.

- Ja w ir pāra, Detje iet no skolas mājup, un viņa paies garām mājām pretējā secībā $N - 1, N - 2, \dots, 0$. Tas ir, Tev vispirms jāielasa un jāizvada b_{N-1} , tad b_{N-2} utt. līdz b_0 .

Kad $w = 1$, ievaddatu vērtības b_0, b_1, \dots, b_{N-1} ir putnu statuju sākotnējie stāvokļi (kas ir arī iemītnieku iecienītie stāvokļi). Kad $w > 1$, ievaddatu vērtības b_0, b_1, \dots, b_{N-1} Tavai programmai būs tādas, kādos stāvokļos statujas nostādīja Tava programma iepriekšējā tās darbināšanas reizē.

Beigās, pēc Tavas programmas pēdējās izpildes reizes b_i vērtībai ir jābūt vienādai ar sākotnējo b_{a_i} vērtību visiem i , citādi Tu saņemsi vērtējumu "Wrong Answer" - kļūdaina atbilde.

Sīkāka informācija.

Ja izpildes laiku $summa W + 1$ Tavas programmas atsevišķajām izpildes reizēm pārsniedz laika limitu, Tava iesūtījuma novērtējums būs "Time Limit Exceeded" - laika limits pārsniegts.

Pārliecinies, ka pēc katras rindas Tu notīri standarta izvadi, citādi Tava programma var saņemt vērtējumu "Time Limit Exceeded" - laika limits pārsniegts. Python valodā izvades notīrīšana notiek automātiski, ja Tu rindu ielasīšanai izmanto `input()`. C++ valodā `cout << endl;` izvada jaunu rindu, kā arī notīra izvadi; ja izmanto `printf`, jālieto `fflush(stdout)`.

Ierobežojumi un vērtēšana

- $2 \leq N \leq 500$.
- Tu drīksti izmantot ne vairāk kā $W \leq 60$ gājienus.

Tavs risinājums tiks testēts ar vairākām testu grupām, kur katra no tām ir noteiktu punktu vērtā. Katrā testu grupā ir vairāki testi. Lai iegūtu punktus testu grupā, ir jāatrisina visi šīs testu grupas testi.

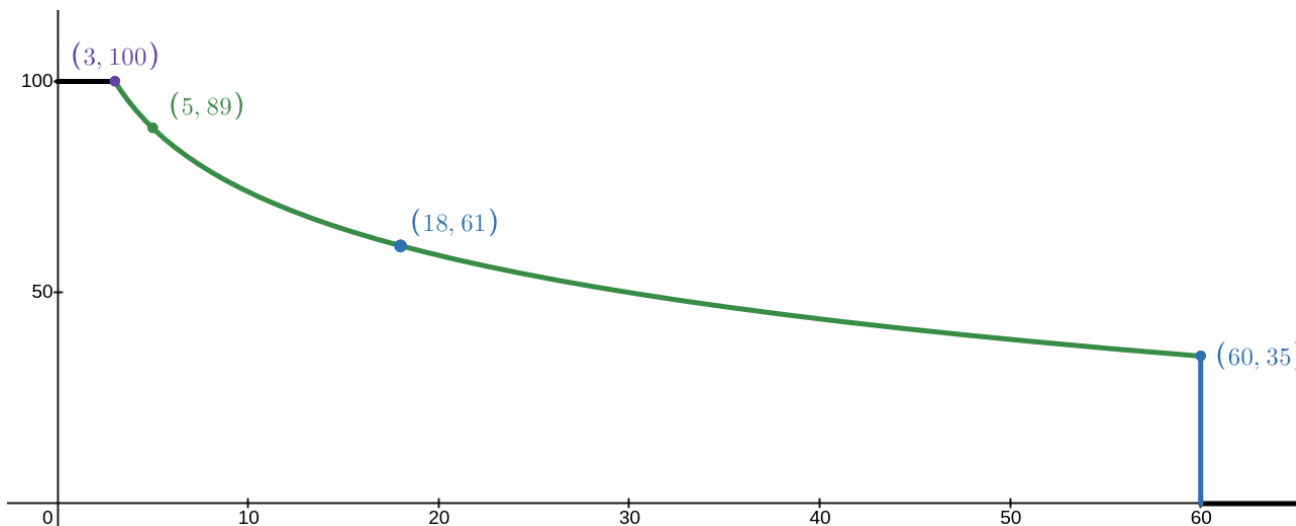
Grupa	Maksimālie punkti	Ierobežojumi
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Bez papildu ierobežojumiem

Katrai testu grupai, kuru Tava programma atrisina pareizi, Tavs punktu skaits tiks aprēķināts šādi:

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

kur S_g ir maksimālie punkti testu grupai un W_g ir maksimālā W vērtība, kas izmantota kādam no testiem šajā testu grupā. Tavi punkti katrai testu grupai tiks noapaļoti līdz tuvākajam veselajam skaitlim.

Tālāk redzamajā grafikā parādīts punktu skaits, kurus Tava programma saņems, ja tā atrisinās visas testu grupas ar vienādu W vērtību, kā funkcija no W . Lai par šo uzdevumu iegūtu 100 punktus, Tev visi testi jāatrisina ar $W \leq 3$.



Testēšanas rīks

Lai atvieglotu Tava risinājuma testēšanu, ir pieejams vienkāršs rīks, ko Tu vari lejupielādēt. Rīks ir atrodams “attachments” sadaļā Kattis uzdevumu vietnes apakšā. Rīka izmantošana nav obligāta. Ņem vērā, ka oficiālā Kattis vērtēšanas programma atšķiras no šī testēšanas rīka.

Lai izmantotu šo rīku, izveido ievaddatu failu, piem., “sample1.in”, kurā vispirms ir skaitlis N , nākamajā rindā ir N skaitļi, kas raksturo permutāciju, un pēdējā rindā ir N biti (0 vai 1), kas norāda putnu statuju sākotnējos stāvokļus. Piemēram:

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

Python programmām, piem., `solution.py` (kas parasti tiek palaista ar `pypy3 solution.py`):

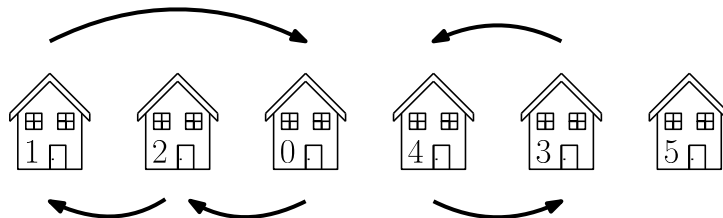
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ programmas vispirms ir jākompilē (piem., ar `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) un tad jāizsauc komada:

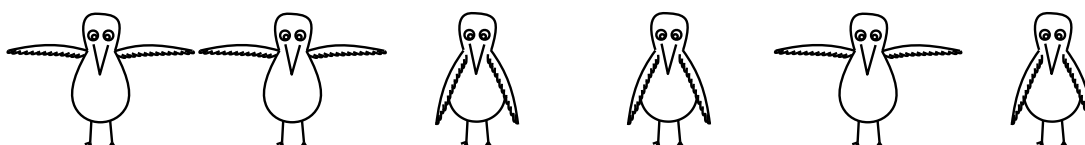
```
python3 testing_tool.py ./solution.out < sample1.in
```

Piemērs

Piemērā dota šāda iemītnieku permutācija mājās:

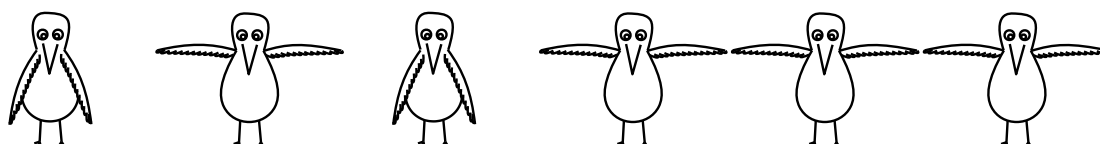


Pirmajā reizē, kad piemēra programma tiek izpildīta (ar $w = 0$), tā izvada $W = 2$, kas nozīmē, ka Detje ies pa ielu divreiz (un programma tiks izpildīta vēl divas reizes). Pirms pirmā gājiena putnu statujas dārzā izskatās šādi:



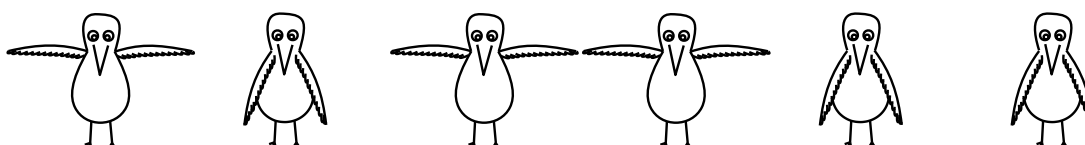
Tad programma tiek izpildīta ar $w = 1$, kas apzīmē Detjes pirmo gājienu. Viņa pa vienam iet gar putnu statujām, sākot no kreisās puses un, iespējams, mainot to stāvokli. Parauga programmai jāizvada i -tās putna statujas stāvoklis pirms tiek ieraudzīta $(i + 1)$ -tā putna statuja.

Pēc Detjes ierašanās skolā putnu statuju stāvokļi izskatās šādi:



Programmas pēdējā izpildes reizē (ar $w = 2$), Detje iet atpakaļ mājās no skolas. Atceries, ka šajā gadījumā viņa ies gar putnu statujām no labās puses uz kreiso un mainīs to stāvokļus šajā secībā! Tas nozīmē, ka viņai jāizlemj i -tās putna statujas stāvoklis pirms viņa ierauga $(i - 1)$ -to putna statuju.

Pēc gājiena pabeigšanas putnu statujas izskatās šādi:



Šī patiešām ir pareizā konfigurācija. Piemēram, 3. putna statuja (ceturtā no kreisās puses) ir ar atplestiem spārniem (tagad $b_3 = 1$), kas ir pareizi, jo 4. iemītņieks pārvāksies uz turieni ($a_3 = 4$) un sākotnēji viņam bija putna statuja ar izplestiem spārniem (sākotnēji $b_4 = 1$).

vērtētāja izvade	Tava izvade
0 6	
1 2 0 4 3 5	
	2

vērtētāja izvade	Tava izvade
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

vērtētāja izvade	Tava izvade
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1