D. Garden Decorations

Tehtävän nimi	Garden Decorations
Aikaraja	7 sekuntia
Muistiraja	1 gigatavu

Joka päivä kulkiessaan kouluun ja kotiin Helen kävelee kadulla, jossa on N taloa, jotka ovat numeroitu 0:sta $N-1$:teen. Tällä hetkellä talossa i asuu henkilö i. Maiseman vaihtamiseksi asukkaat ovat päättäneet vaihtaa taloja keskenään. Henkilö, joka muuttaa taloon i, on henkilö a_i (joka asuu tällä hetkellä talossa a_i).

Jokaisen talon puutarhassa on lintupatsas. Patsailla on kaksi mahdollista tilaa: joko niiden siivet ovat *auki* (ikään kuin lintu lentäisi) tai *kiinni* (ikään kuin se seisoisi maassa). Asukkailla on erittäin vahvat mieltymykset siitä, miltä lintupatsaiden tulee näyttää, ja kieltäytyvät muuttamasta uuteen taloonsa ennen kuin patsas heidän uudessa puutarhassaan näyttää samalta kuin se näytti heidän edellisessä puutarhassaan. Helen haluaa auttaa heitä järjestämään lintupatsaat, jotta he voivat liikkua.

Tätä varten hän tekee seuraavasti: aina kun hän kävelee kadulla (joko matkalla kouluun tai kotiin), hän tarkkailee lintuja, joita hän ohittaa yksitellen ja mahdollisesti säätää joitain patsaita (avaamalla tai sulkemalla niiden siipiä). Koska hänen päivänsä koulussa ja kotona ovat hyvin kiireisiä, **hän ei muista aiemmilla kävelyllään näkemiensä lintujen tilaa**. Onneksi hän on kirjoittanut muistiin listan $a_0, a_1, ..., a_{N-1}$, joten hän tietää, kuka asukas muuttaa minnekin.

Auta Heleniä suunnittelemaan strategia, joka kertoo hänelle, mitä lintuja hänen tulee käsitellä, jotta patsaat sopivat asukkaiden mieltymyksiin. Hän voi kävellä kadun korkeintaan 60 kertaa, mutta korkeamman pistemäärän saavuttamiseksi hänen tulisi kävellä katu mahdollisimman vähän kertoja.

Toteutus

Tämä on usean suorituksen (multirun) tehtävä, mikä tarkoittaa, että ohjelmasi suoritetaan useita kertoja.

Jokaisella suorituskerralla sinun tulee ensin lukea rivi, jossa on kaksi kokonaislukua w ja N, kävelyn indeksi ja talojen lukumäärä. Ohjelmasi ensimmäisellä suorituskerralla $w = 0$, toisella $w = 1$ ja niin

edelleen (lisätietoja selitetään alla).

Toisella syöterivillä on N kokonaislukua $a_0, a_1, ..., a_{N-1}$, mikä tarkoittaa, että henkilö, joka muuttaa taloon i, asuu tällä hetkellä talossa a_i. Luettelo a_i muodostaa *permutaation*: eli jokainen numero välillä 0 - $N-1$ esiintyy täsmälleen kerran a_i -luettelossa. Huomaa, että asukas voi halutessaan olla muuttamatta; eli $a_i = i$ on sallittu.

Asukkaat vaihtavat asuntoa vain kerran. Tämä tarkoittaa, että yksittäisessä testitapauksessa N:n arvo ja a_i-luettelo ovat samat kaikissa ohjelmasi suorituksissa.

Ensimmäinen suorituskerta.

Ohjelmasi ensimmäisellä suorituskerralla $w = 0$. Tällä suorituskerralla sinun tulee yksinkertaisesti tulostaa yksi kokonaisluku W ($0 \leq W \leq 60$), kuinka monta kertaa haluat Helenin kävelevän talojen ohi. Ohjelmasi pitäisi sitten poistua. Tämän jälkeen ohjelmasi suoritetaan uudelleen W kertaa.

Seuraavat suorituskerrat.

Ohjelmasi seuraavassa suorituskerrassa $w = 1$; sen jälkeisessä $w = 2$; ja niin edelleen viimeiseen suoritukseen asti, jossa $w = W$.

Kun olet lukenut w, N ja $a_0, a_1, ..., a_{N-1}$, Helen alkaa kävellä katua pitkin.

- Jos w on pariton, Helen kävelee kotoaan kouluun ja ohittaa talot järjestyksessä $0, 1, ..., N-1$.

  Ohjelmasi pitää nyt lukea rivi b_0, joko 0 (kiinni) tai 1 (auki), talon 0 edessä olevan patsaan nykyinen tila. Kun olet lukenut b_0, sinun tulee tulostaa rivi joko 0 tai 1, uusi arvo, jonka haluat asettaa b_0:lle.

  Sitten ohjelmasi pitää lukea rivi b_1, talon 1 edessä olevan patsaan tila; ja tulosta uusi arvo b_1. Tämä jatkuu jokaiselle N:lle talolle. Kun olet ohittanut viimeisen talon (eli lukenut ja kirjoittanut b_{N-1}), ohjelman tulee poistua.

Huomaa, että ohjelmasi voi lukea seuraavan arvon b_{i+1} vasta sen jälkeen, kun olet kirjoittanut arvon b_i.

- Jos w on parillinen, Helen kävelee koulusta kotiinsa ja ohittaa talot päinvastaisessa järjestyksessä $N-1, N-2, ...0$.

  Prosessi on sama kuin silloin, kun w on pariton, paitsi että aloitat lukemalla ja kirjoittamalla b_{N-1}, sitten b_{N-2} ja niin edelleen, kunnes luet ja kirjoitat arvon b_0.

Kun $w = 1$, syötearvot $b_0, b_1, ..., b_{N-1}$ ovat lintupatsaiden alkuperäinen tila. Kun $w > 1$, ohjelmasi syötearvot $b_0, b_1, ..., b_{N-1}$ ovat samat mitä ohjelman edellinen suorituskerta asetti niille.

Ohjelman viimeisen suorituksen jälkeen b_i:n arvon on oltava sama kuin b_{a_i}:n alkuperäinen arvo kaikille i:lle, muuten saat tuomion Wrong answer.

Yksityiskohdat.

Jos ohjelmasi $W + 1$ eri suorituskertojen suoritusaikojen *summa* ylittää aikarajan, lähetyksesi saa tuloksen Time Limit Exceeded.

Muista huuhdella vakiotuloste jokaisen rivin tulostamisen jälkeen, tai muuten ohjelmasi voi saada tuloksen Time Limit Exceeded. Pythonissa tämä tapahtuu automaattisesti niin kauan kuin käytät `input()` rivien lukemiseen. C++:ssa `cout << endl;` huuhtelee rivinvaihdon lisäksi; jos käytät printf:ää, käytä `fflush(stdout)`.

Rajoitukset ja pisteytykset

- $2 \leq N \leq 500$.
- Voit käyttää enintään $W \leq 60$ kierrosta.

Ratkaisuasi kokeillaan sarjalla testiryhmiä, jotka ovat jokainen tietyn pistemäärän arvoisia. Jokainen testiryhmä sisältää sarjan testitapauksia. Jotta saat testiryhmän pisteet, sinun tulee läpäistä kaikki testitapaukset kyseisessä testiryhmässä.

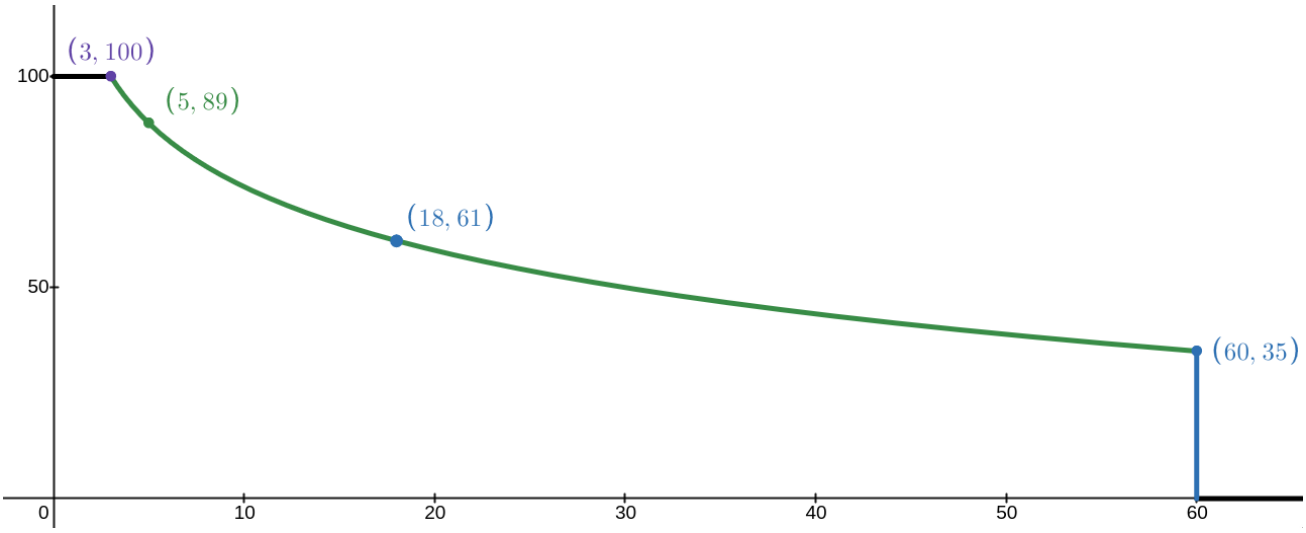
Ryhmä	Maksimipisteet	Rajat
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Ei lisäehtoja

Jokaisesta testiryhmästä, jonka ohjelmasi ratkaisee oikein, saat pisteitä seuraavan kaavan mukaan:

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

missä S_g on testiryhmän enimmäispistemäärä ja W_g on W:n suurin arvo testiryhmän testitapauksissa. Jokaisen testiryhmän pisteet pyöristetään lähimpään kokonaislukuun.

Alla oleva kaavio näyttää pisteiden määrän W:n funktiona, jonka ohjelmasi saa, jos se ratkaisee kaikki testiryhmät samalla W-arvolla. Tarkemmin, saadaksesi 100 pistettä tästä tehtävästä, sinun on ratkaistava jokainen testitapaus $W \leq 3$:lla.



Testaustyökalu

Ratkaisusi testaamisen helpottamiseksi tarjoamme yksinkertaisen työkalun, jonka voit ladata. Katso "attachments" Kattis-ongelmasivun alalaidasta. Työkalun käyttö on vapaaehtoista. Huomaa, että Kattiksen virallinen testijärjestelmä eroaa testaustyökalusta.

Käytä työkalua luomalla syötetiedosto, kuten "sample1.in", jonka tulee alkaa luvulla N. Seuraavalla rivillä tulee olla N lukua, jotka määrittävät permutaation, ja tämän jälkeen toinen rivi N:llä bitillä (0 tai 1), jotka määrittävät lintujen alkutilat. Esimerkiksi:

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

Pythonilla, ohjelma `solution.py` (normaalisti suoritettuna `pypy3 solution.py`):

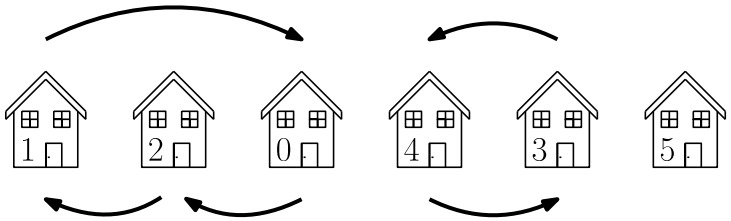
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ ohjelmilla käännä se ensin (esim. `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) ja sitten suorita:

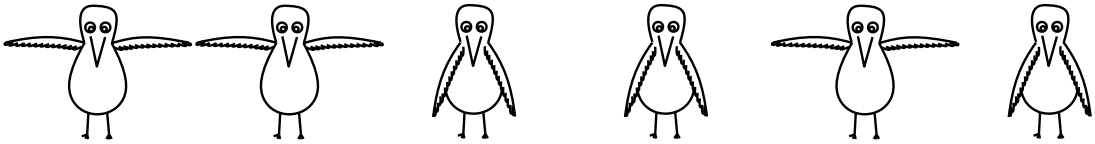
```
python3 testing_tool.py ./solution.out < sample1.in
```

Esimerkit

Esimerkissä annetaan seuraava permutaatio ihmisistä taloissa:

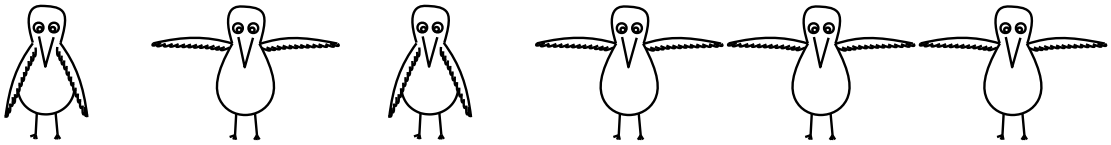


Kun esimerkkiohjelma suoritetaan ensimmäisen kerran ($w = 0$), se tulostaa $W = 2$, mikä tarkoittaa, että Helen kävelee katua pitkin kaksi kertaa (ja ohjelma suoritetaan vielä kaksi kertaa). Ennen ensimmäistä kävelyä puutarhan linnut näyttävät tältä:



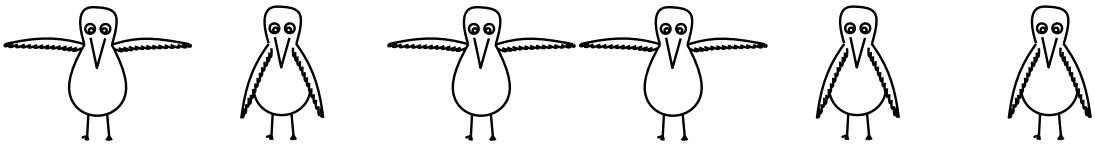
Sitten ohjelma suoritetaan arvolla $w = 1$: tarkoittaen Helenin ensimmäistä kävelyä. Hän käy linnut läpi yksitellen, aloittaen vasemmalta, ja mahdollisesti muuttaa niiden tilaa. Esimerkkiohjelman on tulostettava i:nnen linnun tila ennen kuin näemme $(i + 1)$:nnen linnun.

Kun Helen saapui kouluun, lintujen tila näyttää tältä:



Ohjelman viimeisellä suorituskerralla ($w = 2$), Helen kävelee takaisin koulusta kotiin. Muista, että tässä tapauksessa hän käy linnut läpi oikealta vasemmalle ja käsittelee ne päinvastaisessa järjestyksessä! Tämä tarkoittaa, että hänen on määritettävä i:nnen linnun tila ennen kuin hän näkee $(i - 1)$:nnen linnun.

Kun hän lopettaa kävelynsä, linnut näyttävät tältä:



Tämä on oikea kokoonpano. Esimerkiksi lintupatsas 3 (eli neljäs vasemmalta) on auki (nyt $b_3 = 1$), mikä on oikein, koska henkilö 4 muuttaa sinne ($a_3 = 4$) ja hänellä oli alun perin avoin lintupatsas (alun perin $b_4 = 1$).

testijärjestelmän tuloste	sinun tuloste
0 6	
1 2 0 4 3 5	
	2

testijärjestelmän tuloste	sinun tuloste
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

testijärjestelmän tuloste	sinun tuloste
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1