

D. Aiakaunistused

Ülesande nimi	gardendecoration
Ajapiirang	7 sekundit
Mälupiirang	1 gigabait

Agnes kõnnib iga päev kooli minnes ja koolist koju tulles mööda tänavat, kus on N maja, nummerdatud 0 kuni $N - 1$. Hetkel elab majas i inimene i . Et elu huvitavam oleks, otsustasid majade elanikud omavahel maju vahetada. Majja i kolib inimene a_i (kes elab praegu majas a_i).

Iga maja aias on linnu kuju. Igal kujul on kaks võimalikku asendit: nende tiivad on kas *lahti* (nagu lind lendaks) või *kinni* (nagu lind seisaks maa peal). Majaelanikel on väga tugevad arvamused oma linnukujude välimuse osas. Hetkel on kuju maja i ees elaniku i eelistatud asendis. Elanikud ei ole nõus kolima, kui nende maja ees olev kuju pole nende eelistatud asendis. Agnes tahab aidata neil kujusid kohandada, et elanikud saaks kolida.

Selleks teeb ta järgnevat: iga kord, kui ta mööda tänavat kõnnib (kas kooli minnes või koolist koju tulles), vaatleb ta iga kuju, millest ta möödub, ja soovi korral kohandab selle asendit (kas avab või sulgeb selle tiivad). Kuna ta koolipäevad on väga tihedad, siis **ei mäleta ta kujude, mida ta eelmistel jalutuskäikudel nägi, asendeid**. Õnneks on ta üles kirjutanud nimekirja a_0, a_1, \dots, a_{N-1} , seega ta teab, kes kuhu kolib.

Aita Agnesel välja mõelda strateegia, mis talle ütleks, milliseid kujusid kohandada, et need elanike eelistusega sobiks. Ta saab mööda tänavat kõndida maksimaalselt 60 korda, kuid parema tulemuse saavutamiseks tuleks tal mööda tänavat vähem kordi kõndida.

Implementeerimine

See on mitmekordse väljakutsumisega ülesanne: sinu lahendust joostake mitu korda.

Igal väljakutsel tuleb kõigepealt sisse lugeda üks rida, kus on kaks täisarvu w ja N : käigu järjekorranumber ja majade arv. Programmi esimesel väljakutsumisel on $w = 0$, teisel $w = 1$ jne (allpool on põhjalikumalt seletatud).

Teisel real on N täisarvu a_0, a_1, \dots, a_{N-1} , mis tähendab, et inimene, kes kolib majja i , elab hetkel majas a_i . Arvud a_i moodustavad kokku *permutatsioon*: iga arv vahemikus 0 kuni $N - 1$ esineb nimekirjas täpselt üks kord. Elanik võib ka otsustada mitte kolida, s.t. $a_i = i$ on lubatud.

Elanikud kolivad täpselt üks kord. See tähendab, et ühes testis on N väärtus ja nimekiri a_i igal programmi väljakutsel samad.

Esimene väljakutsumine

Programmi esimesel väljakutsumisel on $w = 0$. Sellel väljakutsel tuleb väljastada vaid üks täisarv W ($0 \leq W \leq 60$): kui mitu korda Agnes majadest mööda kõndima peaks. Programm peaks seejärel töö lõpetama. Seejärel joostakse programmi uuesti W korda.

Järgmised väljakutsumised

Programmi järgmises väljakutses on $w = 1$, ülejäärmises $w = 2$ jne, kuni viimase väljakutseni, kus $w = W$.

Pärast w , N ja a_0, a_1, \dots, a_{N-1} sisse lugemist alustab Agnes jalutuskäiku.

- Kui w on paaritu, siis kõnnib Agnes kodust kooli, ja ta möödub majadest järjekorras $0, 1, \dots, N - 1$.

Sinu programm peaks siis lugema sisse ühe rea, millel on täisarv b_0 , mis kirjeldab maja 0 ees oleva kaju asendit: kas 0 (kinni) või 1 (lahti). Kui oled sisse lugenud arvu b_0 , pead väljastama ühe rea, kus on kas 0 või 1: uus väärtus, milleks b_0 seada.

Siis peaks sinu programm sisse lugema ühe rea, millel on täisarv b_1 , mis on maja 1 ees oleva kaju asend, ja väljastama ühe rea b_1 uue väärtusega. See jätkub kõigi N maja jaoks. Pärast viimasest majast möödumist (s.t. pärast b_{N-1} lugemist ja kirjutamist) **peab su programm töö lõpetama**.

Pane tähele, et su programm võib b_{i+1} väärtuse lugeda alles pärast b_i väärtuse kirjutamist.

- Kui w on paaris, siis kõnnib Agnes koolist koju, ja möödub majadest vastupidises järjekorras: $N - 1, N - 2, \dots, 0$. Protsess on täpselt sama, mis paaritu w jaoks, kuid alustad lugemist ja kirjutamist b_{N-1} -st, siis b_{N-2} , jne kuni b_0 .

$w = 1$ korral on loetavad sisendväärtused b_0, b_1, \dots, b_{N-1} linnukujude algasendid (mis on ka iga elaniku eelistatud asendid). Kui $w > 1$, siis on sisendid b_0, b_1, \dots, b_{N-1} täpselt need, mis sinu programm eelmisel väljakutsel kirjutas.

Lõpuks, pärast programmi viimast väljakutset, peab b_i olema täpselt võrdne b_{a_i} algse väärtusega iga i jaoks. Vastasel korral saab esitus hindeks "Wrong Answer".

Muud märkused

Kui programmi jooksmisaegade *summa* kõigi $W + 1$ väljakutse jaoks kokku ületab ajalimiiti, siis saab esitus hindeks "Time Limit Exceeded".

Pange tähele, et pärast iga rea väljastamist standardväljundisse tuleb puhvrid tühjendada. Vastasel korral võib esitus hindeks saada "Time Limit Exceeded". Pythonis toimub see automaatselt, kui sisendi lugemiseks `input()` funktsiooni kasutada. C++is `cout << endl;` väljastab reavahetuse ja tühjendab puhvrid; kui kasutada `printf`-tüüpi funktsioone, siis peab tegema eraldi `fflush(stdout);`.

Piirangud ja hindamine

- $2 \leq N \leq 500$.
- Võite kasutada maksimaalselt $W \leq 60$.

Sinu lahendust testitakse hulgal testigruppidel, iga neist on väärt mingi arvu punkte. Igas testigrupis on hulk teste. Et saada testigrupi eest punkte, pead läbima kõik gruppi kuuluvad testid.

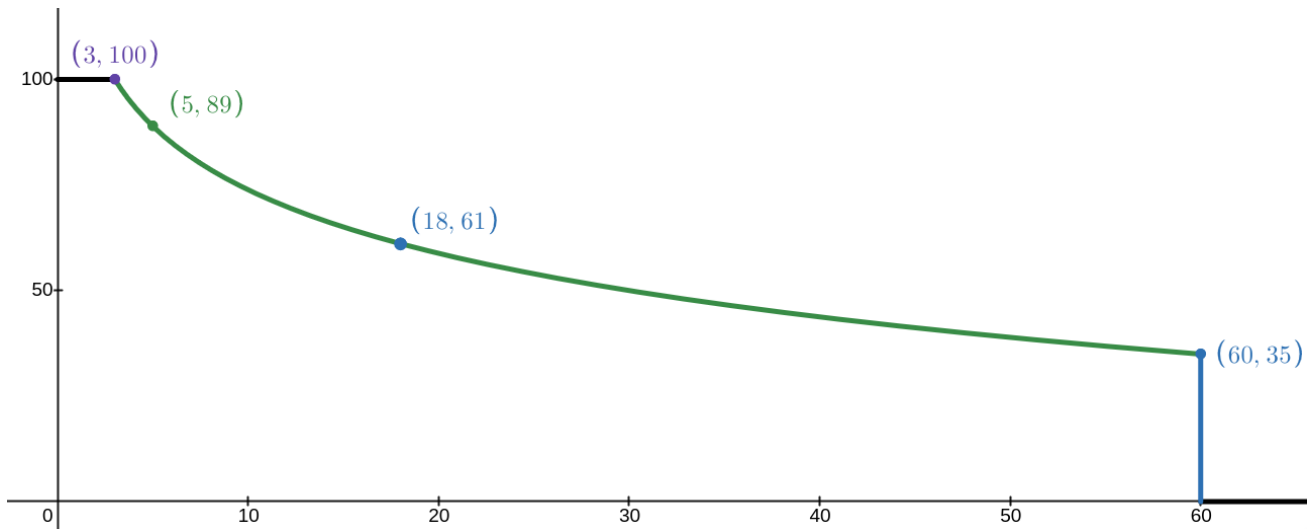
Grupp	Punktid	Piirangud
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Lisapiirangud puuduvad

Iga grupi eest, mille esitatud programm korrektselt lahendab, saab punkte järgneva valemi kohaselt:

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

kus S_g on testigrupi maksimaalne punktisumma ja W_g on kõige suurem W lahendus, mida programm antud grupi testide lahendamisel kasutas. Iga testigrupi punktisumma ümardatakse lähima täisarvuni.

Allolev graafik näitab saadavat punktisummat funktsioonina W -st, lahendus saab nii palju punkte, kui see kõikides testigruppides sama W väärtust kasutab. Spetsiifiliselt tuleb selleks, et 100 punkti saada, lahendada iga test võttes $W \leq 3$.



Testimistöriist

Lahenduse testimise hõlbustamiseks on antud lihtne allalaetav tööriist. Vaata Kattises ülesande lehe allservas "attachments" osa. Tööriista kasutamine ei ole kohustuslik. Pane tähele, et Kattises kasutatav ametlik hindamisprogramm erineb antud tööriistast.

Tööriista kasutamiseks loo sisendfail, näiteks "sample1.in", mis algab arvuga N ja millele järgneb rida N arvuga, mis määrab permutatsiooni a_i , ja veel üks rida N bitiga (0 või 1), mis näitab linnukujude algolekut. Näiteks:

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

Pythoni lahenduse testimiseks, näiteks `solution.py` (mida tavaliselt jookseksid kui `pypy3 solution.py`):

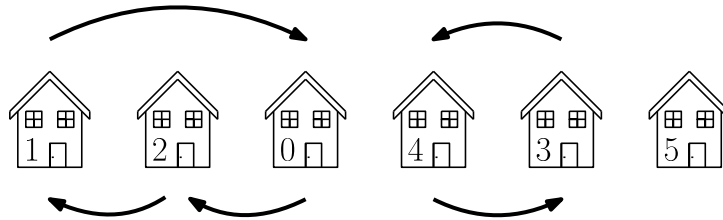
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ lahenduse testimiseks kõigepealt kompileeri lahendus (näiteks käsuga `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) ja seejärel jookse:

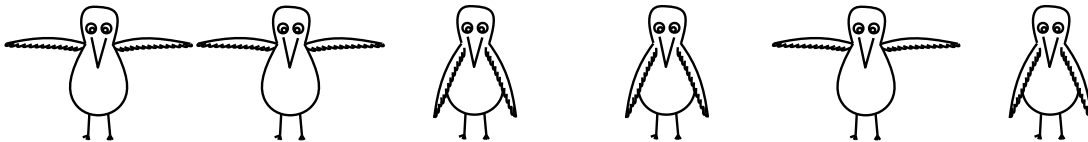
```
python3 testing_tool.py ./solution.out < sample1.in
```

Näide

Näidises on antud järgnev inimeste permutatsioon:

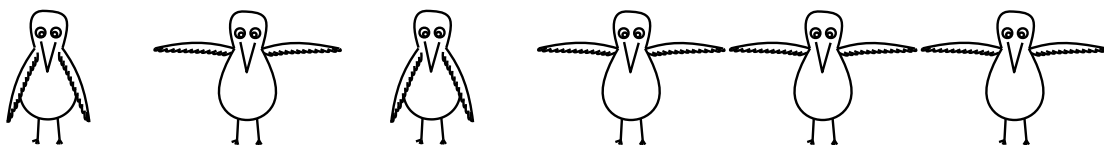


Esimene kord, kui nädisprogramm jookseb (antud $w = 0$), väljastab see $W = 2$, mis tähendab, et Agnes kõnnib mööda tänavat 2 korda (ja programmi joostakse veel 2 korda). Enne esimest jalutuskäiku on linnukujud sellised:



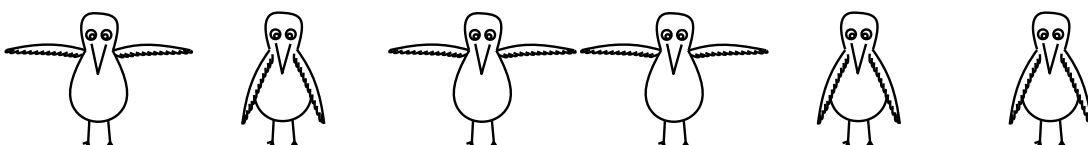
Siis joostakse programmi sisendiga $w = 1$, mis tähistab Agnese esimest jalutuskäiku. Ta kõnnib linnukujudest ükshaaval mööda (alustades vasakult) ja vajadusel kohandab neid. Programm peab väljastama kuju i oleku enne, kui kuju $(i + 1)$ olekut näeb.

Kui Agnes üksinda koolimajja jõudis, olid ~~tunnid juba alanud~~ kujude olekud järgnevad:



Viimasel programmi väljakutsumisel (kus $w = 2$) kõnnib Agnes koolist tagasi koju. Pane tähele, et siis näeb ta kujusid alates paremalt ja minnes vasakule! See tähendab, et kuju i olek tuleb väljastada enne kuju $(i - 1)$ oleku nägemist.

Pärast jalutuskäiku näevad kujud välja sellised:



Mis on tõepoolest õige paigutus. Näiteks linnukuju 3 (s.t. vasakult neljas) on lahti (ehk $b_3 = 1$), mis on õige, kuna inimene 4 kolib sinna ($a_3 = 4$) ja tal oli algselt laheline linnukuju (algselt $b_4 = 1$).

hindamisprogrammi väljund	lahenduse väljund
0 6	
1 2 0 4 3 5	
	2

hindamisprogrammi väljund	lahenduse väljund
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

hindamisprogrammi väljund	lahenduse väljund
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1