

D. Kerti díszpintyek

Feladat neve	Garden Decorations
Időkorlát	7 másodperc
Memóriakorlát	1 gigabyte

Minden nap az iskolába menet és onnan hazafelé Detje végigsétál egy utcán. Ebben az utcában N ház található, melyeket Detje otthonától az iskola felé haladva 0-tól $N - 1$ -ig sorszámozunk. A környezetváltozás kedvéért a házak lakói elhatározzák, hogy lakóhelyet cserélnek egymással: az i sorszámú házba az a_i sorszámú ház jelenlegi lakója költözik majd be.

Minden ház kertjében egy pintyet ábrázoló szobor található. Ezek a díszpintyek kétféle (tetszőlegesen sokszor megváltoztatható) állapotban lehetnek: a szárnyaik vagy *kitárva* (mintha a madár repülne), vagy *összecsukva* (mintha a madár a földön állna) lehetnek. A lakók nagyon kényesek a díszpintyeik állapotára, így csakis akkor hajlandóak átköltözni az új házukba, ha annak kertjében a díszpinty ugyanolyan állapotú, mint a jelenlegi kertjükben lévő. Detje segíteni szeretne nekik a költözésben a díszpintyek állapotának megfelelő megváltoztatásával.

Ehhez a következő szerint jár el: bármikor, amikor végighalad az utcán (akár az iskolába menet, akár hazafelé), megfigyeli a díszpinty állapotát abban a kertben, ami mellett épp elhalad, és tetszése szerint megváltoztathatja azt (kinyithatja vagy összecsukhatja a szárnyait). Mivel Detje nagyon elfoglalt, így sajnos az egyes végighaladások során már *nem emlékszik*, hogy korábban milyen állapotban találta ott a díszpintyeket. Szerencsére a költözéseket megadó a_0, a_1, \dots, a_{N-1} értékeket leírta, így azokat nem felejtí el.

Segíts Detje-nek olyan stratégiát választani, amely szerint változtatva az egyes díszpintyek állapotait végül mindegyik eléri a költöző lakók által kívánt állapotot. Ehhez legfeljebb 60 alkalommal sétálhat végig az utcán, de a magasabb pontszám eléréséhez minél kevesebb sétával kell megoldanod a feladatot.

Megvalósítás

A megoldásod értékelése során minden tesztet értékelésekor a programod *több alkalommal kerül futtatásra*.

Minden futtatás során a bemenet első sorából beolvashatja a program a jelenlegi futtatás w sorszámát, valamint a házak N számát. Az első futtatáskor $w = 0$, aztán másodjára $w = 1$, és így tovább (ennek részleteit lentebb ismertetjük).

A második sorban N darab egész szám található, az a_0, a_1, \dots, a_{N-1} értékek, ami azt jelenti, hogy az i sorszámú házba az a_i sorszámú ház jelenlegi lakója szeretne átköltözni. Az a_i számok *permutációt* alkotnak, azaz minden 0 és $N - 1$ közti egész érték pontosan egyszer szerepel köztük. Lehetséges, hogy valaki nem fog költözni, ebben az esetben $i = a_i$ teljesül.

Egy adott tesztesetben a házak száma és az a_i értékek listája rögzített a programod összes futása során.

Első futtatás

A programod első futtatása során $w = 0$. Ennél a futtatásnál egyetlen egész értéket kell a kimenetre írni, azt a W ($0 \leq W \leq 60$) értéket, ahány alkalommal Detje-nek végig kell sétálnia a házak mellett. Ezután a programodnak be kell fejeznie a futását. A programodat további W alkalommal fogják lefuttatni.

További futtatások

A programod következő futtatása során $w = 1$, azt követően $w = 2$, és így tovább, míg az utolsó futtatás során $w = W$.

Miután a programod beolvasta a w , N és az a_0, a_1, \dots, a_{N-1} értékeket, Detje sétáját az utcában az alábbiak szerint szimuláljuk.

- Ha w páratlan, Detje otthonról az iskola felé sétál, tehát a házakat a $0, 1, \dots, N - 1$ sorrendben látogatja meg.

A programodnak először be kell olvasnia a b_0 bemeneti értéket, ami a 0 számú ház kertjében található díszpinty állapotát írja le: értéke vagy 0 (összecsukott), vagy 1 (kitárt). Ezután egy sort kell kiírnia a kimenetre, mely a díszpinty új állapotát írja le a 0 vagy 1 számok valamelyikével.

Következőnek a b_1 értéket kell beolvasni, mely az 1 számú ház kertjében található díszpinty jelenlegi állapota, majd kiírni az új állapotot leíró értéket. Ezt folytatjuk az N ház mindegyikére. Az utolsó (azaz $N - 1$ sorszámú) ház feldolgozása után a programodnak be kell fejeznie a futását.

Vedd figyelembe, hogy a programod a soron következő b_{i+1} értéket csak azt követően olvashatja be, ha már kiírta a kimenetre az új b_i értéket.

- Ha w páros, Detje az iskolából sétál hazafelé, tehát a házakat az $N - 1, N - 2, \dots, 0$ sorrendben látogatja meg.

A folyamat ekkor megegyezik a páratlan esetben leírtakkal, azzal a különbséggel, hogy a b_{N-1} érték olvasásával és kiírásával kezd a programod, ezt követi a b_{N-2} érték, és így tovább, egészen b_0 értékig.

A $w = 1$ esetben beolvasott b_0, b_1, \dots, b_{N-1} értékek a kertekben található díszpintyek kezdeti állapotait adják meg. A $w > 1$ esetekben a beolvasott b_0, b_1, \dots, b_{N-1} értékek megegyeznek a programod által az előző futtatás során megadottakkal.

Végül, a programod utolsó futtatását követően a b_i értékek mindegyikének azonosnak kell lennie az eredeti (az első futtatás előtti) b_{a_i} értékekkel. Ha ez nem teljesül, akkor a megoldásod `Wrong Answer` értékelést kap.

Egyebek

Amennyiben a programod $W + 1$ futtatása során a futási idők összege átlépi az időkorlátot, a megoldásod a `Time Limit Exceeded` értékelést kapja.

Bizonyosodj meg róla, hogy kimenetre küldött adatok ténylegesen kiírása kerülnek, különben a megoldásod a `Time Limit Exceeded` értékelést kaphatja.

A Python programozási nyelv esetén ez automatikusan megtörténik, ha az `input()` parancsot használod a bemenet olvasására.

A C++ nyelv esetén a `cout << endl;` paranccsal kiírt sorvéggel lehet ezt elérni; a `printf` kiíró parancs használata esetén pedig a `fflush(stdout)` parancs használható.

Korlátok és pontozás

- $2 \leq N \leq 500$.
- Legfeljebb $W \leq 60$ futtatás lehetséges.

A megoldásodat különböző tesztcsoportokon ellenőrzik, ahol minden tesztcsoportnak önálló pontértéke van. Minden tesztcsoport több tesztesetet tartalmaz. Egy tesztcsoport pontjainak megszerzéséhez a programodnak a tesztcsoport összes tesztesetét helyesen kell megoldania.

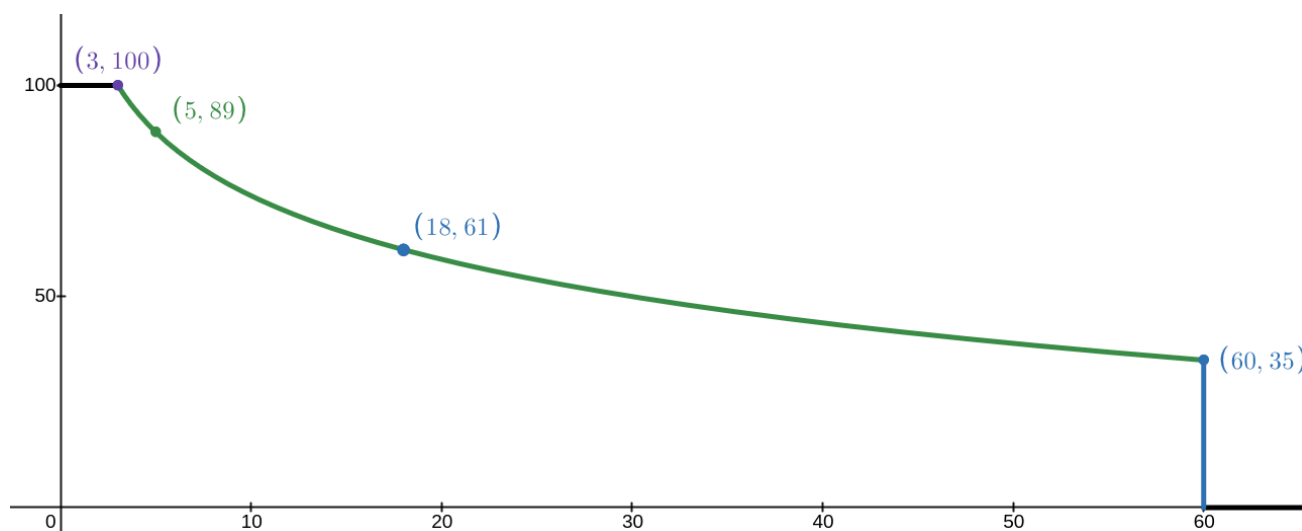
Tesztcsoport	Pontszám	Korlátok
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Nincsenek további korlátok

Minden helyesen megoldott tesztcsoportra a pontszámot az alábbi formula alapján számolják ki:

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

ahol S_g a tesztcsoportra szereshető pontszám, W_g pedig a legnagyobb W érték amit a programod a tesztcsoport tesztéseinek megoldása során használt. A megszerzett pontjaidat minden tesztcsoportra a legközelebbi egész értékre kerekítjük.

Az alábbi ábra a megszereshető pontokat ábrázolja a W érték függvényében, feltéve, hogy a programod helyesen megoldja az összes tesztcsoportot, és a megoldás során a W értéke mindig azonos. Speciálisan, a maximális 100 pont eléréséhez minden tesztet $W \leq 3$ teljesítésével kell megoldani.



Segédprogram

A megoldásod elkészítéséhez letölthetsz egy egyszerű segédprogramot, mely a feladat Kattis rendszerben elérhető oldalának alján, az "Attachments" résznél található. Ennek a programnak a használata nem kötelező, és az értékelőrendszer által használt kiértékelés eltér a segédprogram működésétől.

A segédprogram használatához hozz létre egy bemeneti állományt, például a "sample1.in"-t. Az állomány első sora az N értéket tartalmazza. A második sor a költözéseket megadó permutáció a_i értékeit tartalmazza, a harmadik sor pedig a díszpintyek kezdeti állapotát leíró 0 – 1 értékeket. A lentebb található példát az alábbi módon lehet így megadni:

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

Ezután a futtatáshoz Python nyelvű megoldás, például `solution.py` esetén (melyet normál esetben `pypy3 solution.py` paranccsal futtatnál) a következő parancssori utasítás használható:

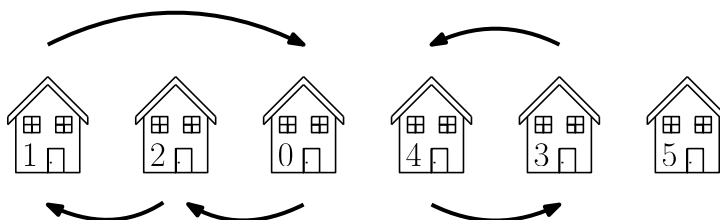
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

A C++ nyelv esetén a programot először le kell fordítani (például `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out` utasítással) és aztán az alábbi módon futtatni:

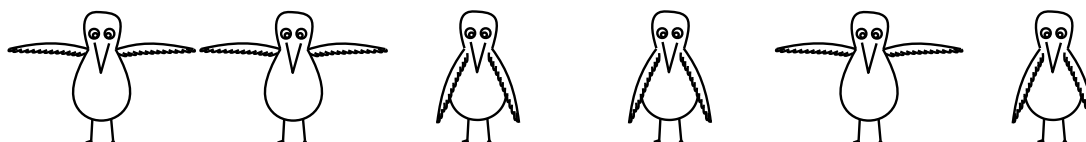
```
python3 testing_tool.py ./solution.out < sample1.in
```

Példa

A példában a következő permutáció szerint költöznek a lakók:

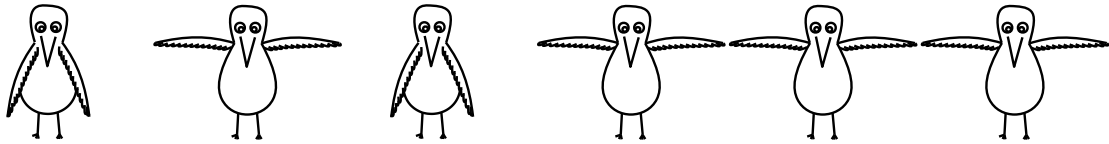


Első alkalommal a mintaként leírt megoldás ($w = 0$ bemenetet olvasva) kiírja a $W = 2$ értéket, ami azt jelenti, hogy Detje két alkalommal fog végigsétálni az utcán. Ezt két további futtatás fogja követni. Az első futtatás előtt a díszpintyek az alábbi állapotokban vannak a házak kertjében:



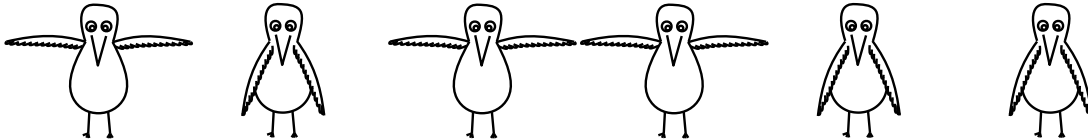
A programot most a $w = 1$ bemenettel futtatjuk, mely Detje első sétájára utal. Az ábra szerint balról jobbra halad végig a díszpintyeken, és tetszés szerint megváltoztathatja az állapotaikat. A programnak ki kell írnia az i -edik díszpinty új állapotát, mielőtt beolvassa az $(i + 1)$ -edik díszpinty jelenlegi állapotát.

Amikor Detje megérkezik az iskolába, a madarak az alábbi állapotban vannak:



Ezt követi a program utolsó futtatása ($w = 2$ értékkel). Detje most hazafelé tart az iskolából: ez azt jelenti, hogy az ábrán jobbról balra halad végig a díszpintyeken, tehát az előzővel ellentétes sorrendben! Formálisan most a programnak ki kell írnia az i -edik díszpinty új állapotát, mielőtt beolvassa az $(i - 1)$ -edik madár jelenlegi állapotát.

A sétája végén a díszpintyek állapotai az alábbiak:



Látható, hogy most az összes madár a kívánt állapotban van. Például a 3-as számú díszpinty (balról a negyedik az ábrán) szárnya ki van tárva ($b_3 = 1$), ami megfelelő, hiszen a 4-es számú ház lakója fog oda költözni ($a_3 = 4$) és az ő kertjében eredetileg a díszpinty szárnyai kitárt állapotban voltak ($b_4 = 1$).

grader output	your output
0 6	
1 2 0 4 3 5	
	2

grader output	your output
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

grader output	your output
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1