

D. Декорации на Дворови

Име на задачата	Декорации на Дворови
Временско ограничување	7 секунди
Мемориско ограничување	1 гигабајт

Секој ден кога оди на училиште и назад дома, Јована оди по улица со N куќи, нумерирани со целите броеви од 0 до $N - 1$. Во моментот, куќата i е населена од лицето i . За да се случува нешто, жителите решиле да ги сменат куќите меѓусебно. Лицето кое ќе се пресели во куќата i е лицето a_i (кое моментално живее во куќата a_i).

Секоја куќа има статуа на птица во дворот. Статуите имаат две можни состојби; крилјата им се *отворени* (како што изгледа птицата кога лета) или *затворени* (како што изгледа кога стои на земјата). Жителите имаат многу силни преференци за тоа како треба да изгледаат нивните статуи на птици. Моментално, птицата пред куќата i е во преферираната состојба за жителот i . Даден жител одбива да се пресели во куќа доколку нејзината статуа не е поставена на преферираната (за него) состојба. Јована сака да им помогне да ги уредат статуите на птиците, за да можат да се преселат.

За оваа цел, таа го прави следново: секогаш кога оди долж улицата (или на патот до училиштето, или пак назад дома), таа ги набљудува птиците што ги поминува една по една и потенцијално прилагодува некои од статуите (отворајќи или затворајќи им ги крилјата). Бидејќи Јована многу учи и е доста занесена, **таа не се сеќава на состојбите на птиците што ги видела на нејзините претходни прошетки**. За среќа, Јована ја има запишано листата a_0, a_1, \dots, a_{N-1} , така што знае кој жител каде се преселува.

Помогнете ѝ на Јована да дизајнира стратегија која ќе ѝ каже кои птици да ги промени за да ги прилагоди статуите според преференциите на жителите. Таа може да помине по улицата најмногу 60 пати, но за да постигне повисок резултат треба да помине по улицата помалку пати.

Имплементација

Ова е проблем со повеќе извршувања, што значи дека вашата програма ќе биде извршена повеќе пати.

Во секое извршување, прво треба да прочитате линија со два цели броја w и N , редниот број (индексот) на прошетката и бројот на куќи. Во првото извршување на вашата програма $w = 0$, во второто $w = 1$, и така натаму (повеќе детали се објаснети подолу).

Во втората линија од влезот има N цели броеви a_0, a_1, \dots, a_{N-1} , што означува дека личноста која ќе се пресели во куќата i во моментот живее во куќата a_i . a_i -овците формираат *пермутација*: т.е., секој од броевите од 0 до $N - 1$ се појавува точно еднаш во листата на a_i -овци. Имајте предвид дека даден жител може да избере да не се пресели; тоа значи дека $a_i = i$ е дозволено.

Жителите сменуваат куќа само еднаш. Ова значи дека за фиксен тест случај, вредноста на N и листата на a_i -овци ќе бидат исти во сите извршувања на вашата програма.

Прво Извршување.

За првото извршување на вашата програма, $w = 0$. Во ова извршување, едноставно треба да отпечатите еден цел број W ($0 \leq W \leq 60$), бројот на прошетки кои сакате Јована да ги направи покрај куќите. Вашата програма потоа треба да заврши. Потоа, вашата програма ќе биде извршена повторно уште W пати.

Следни Извршувања.

Во следното извршување на вашата програма, $w = 1$; во следното потоа $w = 2$; и така натаму, до финалното извршување каде $w = W$.

Откако ќе ги прочитате w , N и a_0, a_1, \dots, a_{N-1} , Јована започнува да оди по улицата.

- Ако w е непарен, Јована оди од нејзиниот дом кон училиштето, и ќе помине покрај куќите во редоследот: $0, 1, \dots, N - 1$.

Вашата програма сега треба да прочита ред со b_0 кое може да е или 0 (затворени) или 1 (отворени), моменталната состојба на крилата на статуата пред куќата 0. Откако ќе го прочитате b_0 , треба да отпечатите линија со 0 или 1, новата вредност која сакате да ја поставите за b_0 .

Потоа вашата програма треба да прочита ред со b_1 , состојбата на статуата пред куќата 1; и да ја отпечати новата вредност за b_1 . Ова продолжува за секоја од N -те куќи. Откако Јована ќе ја помине последната куќа (т.е., ќе го прочитате и отпечатите b_{N-1}), **вашата програма треба да заврши.**

Имајте предвид дека вашата програма може да ја прочита следната вредност b_{i+1} само откако ќе ја отпечати новата вредност на b_i .

- Ако w е парен, Јована оди од училиштето до нејзиниот дом, и ќе помине покрај куќите во обратен редослед $N - 1, N - 2, \dots, 0$.

Тоа значи дека започнувате со читање и печатење на b_{N-1} , па потоа b_{N-2} , и така натаму до b_0 .

Кога $w = 1$, влезните вредности b_0, b_1, \dots, b_{N-1} се оригиналните состојби на статуите на птиците (коишто се исто така и преферираните состојби на жителите). Кога $w > 1$, влезните вредности b_0, b_1, \dots, b_{N-1} во вашата програма ќе бидат оние на кои ги поставило претходното извршување на вашата програма.

На крај, по последното извршување на вашата програма, вредноста на b_i мора да биде еднаква на оригиналната вредност на b_{a_i} за сите i , инаку ќе добиете оценка *Wrong answer*.

Детали.

Ако збирот од времињата на извршување на $W + 1$ засебни извршувања на вашата програма го надмине временскиот лимит, тоа ќе биде оценето со *Time Limit Exceeded*.

Осигурете се дека сте направиле `flush` на `standard output` после принтање на секој ред, оти во противно вашата програма може да добие *Time Limit Exceeded*. Во Python, ова е автоматски ако користите `input()` за читање. Во C++, `cout << endl;` flush-ува покрај тоа што принта и `newline`; но ако користите `printf`, употребете `fflush(stdout)`.

Ограничувања и бодување

- $2 \leq N \leq 500$.
- Може да користите најмногу $W \leq 60$ рунди.

Вашето решение ќе биде тестирано на множество од тест групи, пришто секоја носи одреден број на поени. Секоја тест група содржи множество од тест случаи. За да ги освоите поените за дадена тест група, мора да ги решите сите тест случаи во таа тест група.

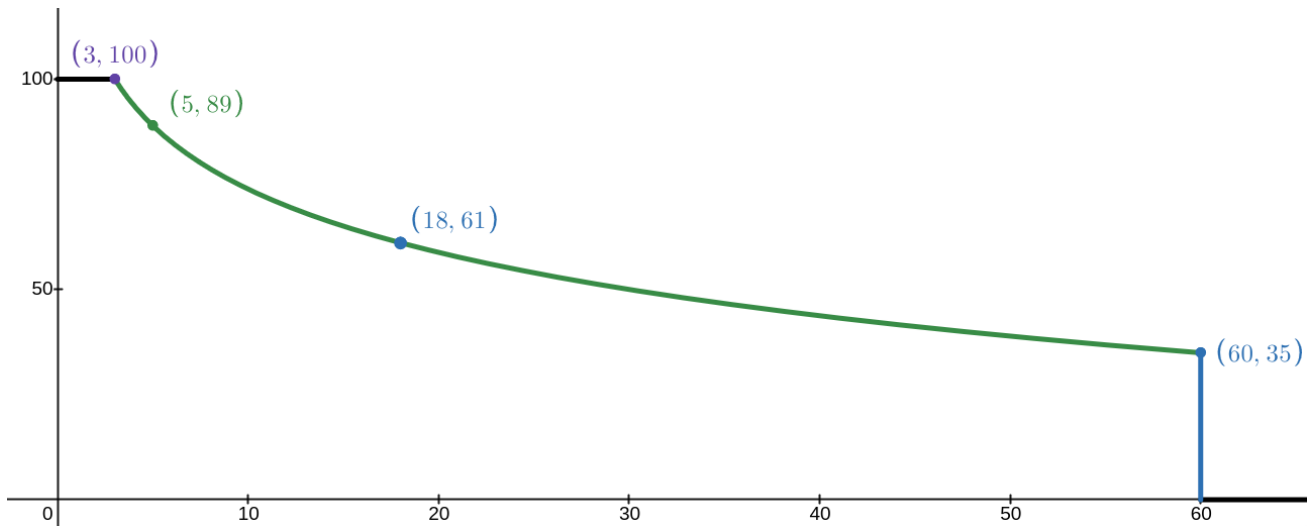
Група	Мах поени	Ограничувања
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Без дополнителни ограничувања

За секоја тест група што точно ќе ја решите ќе добиете поени според следната формула:

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

каде S_g е максималниот резултат за тест групата, и W_g е максималната вредност за W искористен во некој тест случај од групата. Резултатот за тест групата ќе биде заокружен до најблискиот цел број.

Долната слика покажува колку поени се добиваат во зависност од големината на W , ако сите тест групи се решат со истата вредност на W . Конкретно, за 100 поени, треба да го решите секој тест случај со $W \leq 3$.



Алатка за Тестирање

За да го олесниме тестирањето на вашето решение, ви нудиме едноставна алатка која можете да ја преземете. Погледнете ги „дополнителните фајлови (attachments)“ на дното на страницата на проблемот на системот Kattis. Алатката е опционална за користење. Имајте предвид дека официјалниот оценувач на Kattis е различен од тест алатката.

За да ја искористите алатката, создадете влезен фајл, како што е „sample1.in“, кој треба да започнува со бројот N , следен од линија со N броеви кои ја специфицираат пермутацијата, и уште една линија со N бита (0 или 1) кои ги специфицираат почетните состојби на птиците. На пример:

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

За Python програми, да речеме `solution.py` (што вообичаено се извршуваат со `python3 solution.py`):

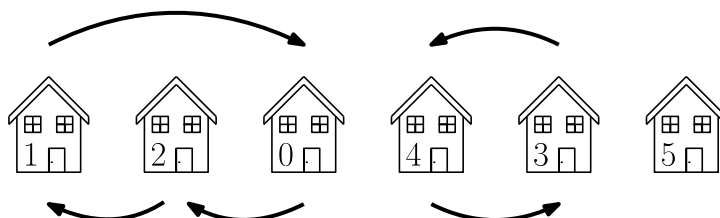
```
python3 testing_tool.py python3 solution.py < sample1.in
```

За C++ програми, прво компајлирајте (e.g. with `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) а потоа извршете:

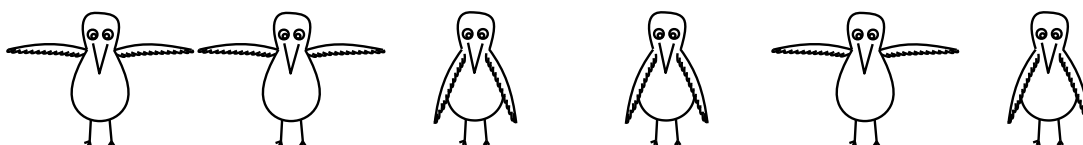
```
python3 testing_tool.py ./solution.out < sample1.in
```

Пример

Во примерот, ни е дадена следната пермутација на луѓе во куќите:

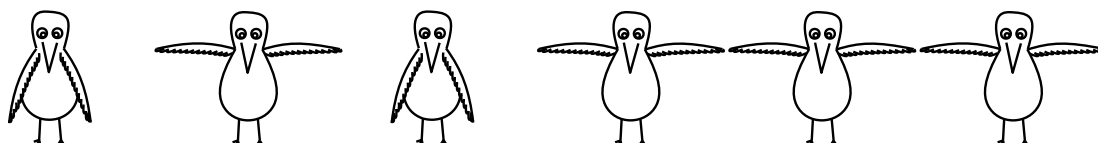


Првиот пат кога програмата се извршува (со $w = 0$), таа враќа $W = 2$, што значи дека Јована ќе оди по улицата два пати (и програмата ќе се изврши уште два пати). Пред првата прошетка, птиците во дворовите изгледаат вака:



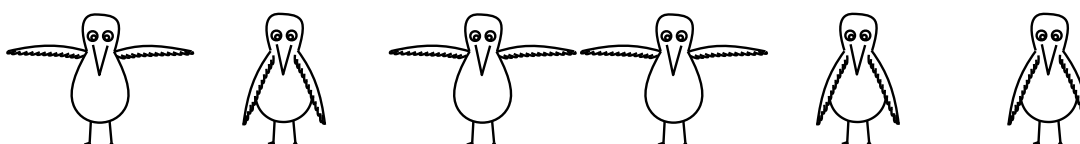
Потоа програмата се извршува со $w = 1$, што ја означува првата прошетка на Јована. Таа поминува покрај птиците една по една, почнувајќи од лево, и можеби ја менува нивната состојба. Програмата треба да ја врати состојбата на i -тата птица пред да ја видиме $(i + 1)$ -тата птица.

Откако Јована ќе стигне во училиште, состојбите на птиците изгледаат вака:



Во финалното извршување на програмата (со $w = 2$), Јована оди назад дома од училиште. Запомнете дека во овој случај, таа ќе поминува низ птиците од десно кон лево и ќе ги обработува во овој редослед! Ова значи дека таа треба да ја одреди состојбата на i -тата птица пред да ја види $(i - 1)$ -тата птица.

Откако ќе ја заврши својата прошетка, птиците сега изгледаат вака:



Всушност, ова е точната конфигурација. На пример, птицата статуа 3 (т.е. четвртата од лево) е отворена (сега $b_3 = 1$), што е точно бидејќи личноста 4 ќе се движи таму ($a_3 = 4$) и таа оригинално имала отворена птица статуа (оригинално $b_4 = 1$).

излез од оценувачот	ваш излез
0 6	
1 2 0 4 3 5	
	2

излез од оценувачот	ваш излез
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

излез од оценувачот	ваш излез
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1