

## D. Garden Decorations

Nafn Verkefnis	Garden Decorations
Tímatakmark	7 sekúndur
Minnistakmark	1 gígabæt

Hvern einasta dag sem Arnar fer í skólann og aftur heim þá gengur hann meðfram götu með  $N$  húsum sem eru númeruð frá 0 upp í  $N - 1$ . Eins og er þá býr manneskja  $i$  í húsi  $i$ . Til að breyta um umhverfi hafa íbúar ákveðið að skipta um hús við hvern annan. Sá sem flytur inn í hús  $i$  er manneskja  $a_i$  (sem býr í húsi  $a_i$ ).

Í hverju húsi er eins konar fuglastytta í garðinum. Stytturarnar hafa tvær mögulegar stöður, annað hvort með vængina **opna** (eins og fuglinn sé að fljúga) eða **lokaða** (eins og fuglinn standi á jörðinni). Íbúar hafa sterkar skoðanir um hvernig fuglastytturarnar þeirra eiga að líta út og neita að flytja í nýja húsið þeirra áður en stytta í nýja garðinum lítur eins út og hún er í þeirra garði. Arnar langar að hjálpa þeim að raða fuglastyttunum svo þau geta farið að flytja.

Til þess gerir hann eftirfarandi: í hvert skipti sem hann gengur meðfram götunni (hvort sem hann er á leiðinni í skólann eða heim) þá fylgist hann með fuglunum sem hann rekst á og mögulega lagar nokkrar stytur (með því að opna eða loka vængjunum þeirra). Þar sem hann er mjög upptekinn í skólanum og heima **þá man hann ekki hvernig fuglarnir voru á fyrri gönguferðum sínum**. Sem betur fer þá skrifaði hann niður lista  $a_0, a_1, \dots, a_{N-1}$ , svo hann man hvaða íbúi er að flytja hvert.

Hjálpaðu Arnari að hanna aðferð sem segir honum hvaða fuglum hann á að breyta til að laga stytturnar eftir smekk íbúana. Hann getur gengið meðfram götunni í mesta lagi 60 sinnum en til að ná hærri fjölda stiga þá ætti hann að ganga sjaldnar meðfram götunni.

### Útfærsla

Þetta er fjölkeyrslu verkefni, sem þýðir að forritið verður keyrt mörgum sinnum.

Í sérhverri keyrslu skaltu fyrst lesa eina línu með tveimur heiltölum  $w$  og  $N$ , númer göngu Arnars og fjöldi húsa. Í fyrstu keyrslunni verður  $w = 0$ , í næstu  $w = 1$  og svo framvegis (nánari smáatriðum er lýst að neðan).

Á annarri línu inntaksins eru  $N$  heiltölur  $a_0, a_1, \dots, a_{N-1}$ , sem þýðir að íbúinn sem mun flytja í hús  $i$  býr í húsi  $a_i$  eins og er. Runan táknar umröðun, sem þýðir að sérhver tala á bilinu 0 upp í  $N - 1$  kemur fyrir nákvæmlega einu sinni. Athugaðu að íbúi má sleppa því að flytja, það er  $a_i = i$  er leyfilegt.

Íbúarnir skipta einungis einu sinni um hús. Það þýðir að fyrir ákveðið prufutilvik er listinn af  $a_i$  gildunum hinn sami fyrir allar keyrslur forritsins.

## Fyrsta Keyrsla

Í fyrstu keyrslu forritsins er  $w = 0$ . Í þessari keyrslu áttu einfaldlega að skrifa út eina heiltölu  $W$ , þar sem  $0 \leq W \leq 60$ , sem er fjöldi skipta sem Arnar mun ganga framhjá húsunum. Forritið þitt skal svo hætta keyrslu. Eftir þetta verður forritið þitt keyrt  $W$  sinnum í viðbót.

## Síðari Keyrslur

Í næstu keyrslu forritsins mun  $w = 1$ ; í þar næstu mun  $w = 2$ ; og svo framvegis þangað til lokakeyrslan er keyrð með  $w = W$ .

Eftir að þú hefur lesið inn  $w$ ,  $N$  og  $a_0, a_1, \dots, a_{N-1}$  byrjar Arnar að ganga meðfram götunni.

- Ef  $w$  er oddatala mun Arnar ganga frá heimili sínu í áttina að skólanum og mun því fara fram hjá húsunum í röðinni  $0, 1, \dots, N - 1$ .

Forritið þitt skal nú lesa línu með  $b_0$ , sem er annaðhvort 0 (lokaðir vængir) eða 1 (opnir vængir), og táknar núverandi stöðu styttnar fyrir framan hús 0. Eftir að hafa lesið inn  $b_0$  skaltu skrifa út línu með annaðhvort 0 eða 1 sem táknar nýja gildið sem þú vilt gefa  $b_0$ .

Því næst skal forritið þitt lesa línu með  $b_1$ , stöðu styttnar fyrir framan hús 1; og skrifa út nýtt gildi á  $b_1$ . Þetta heldur áfram fyrir öll  $N$  húsin. Eftir að þú ferð fram hjá síðasta húsinu (semsagt þú lest inn og skrifar út  $b_{N-1}$ ) skal forritið þitt hætta keyrslu.

*Athugaðu að forritið þitt getur aðeins lesið næsta gildi  $b_{i+1}$  eftir að hafa skrifað gildið á  $b_i$ .*

- Ef  $w$  er slétt tala mun Arnar ganga frá skólanum í átt að heimili sínu og fer því fram hjá húsunum í öfugu röðinni  $N - 1, N - 2, \dots, 0$ . Semsagt þú byrjar að lesa og skrifa  $b_{N-1}$ , svo  $b_{N-2}$ , þar til þú kemur að  $b_0$ .

Þegar  $w = 1$  eru inntaksgildin á  $b_0, b_1, \dots, b_{N-1}$  upprunalegu stöður fuglastyttnanna. Þegar  $w > 1$  eru inntaksgildin á  $b_0, b_1, \dots, b_{N-1}$  gildin sem forritið þitt stillti þau sem í keyrslunni á undan.

Í lokin eftir síðustu keyrslu forrits þíns þarf  $b_i$  að vera jafngilt upprunalega gildinu á  $b_{a_i}$  fyrir öll  $i$ , annars færðu dómsúrskurðinn Wrong Answer.

## Aukaatriði

Ef *summan* af keyrslutíma  $W + 1$  mismunandi keyrslunanna fer yfir tímatakörkin munu skilin þín fá dómsúrskurðinn Time Limit Exceeded.

Eftir sérhverja línu sem þú skrifar út vertu viss um að sturta staðalúttaki, annars gæti forritið þitt fengið dómsúrskurðinn Time Limit Exceeded. Í Python gerist þetta sjálfkrafa ef þú notar `input()`. Í C++ sturtar `cout << endl;` sjálfkrafa og skrifar út nýja línu; ef þú notar `printf` skaltu nota `fflush(stdout)` til að sturta úttaki.

## Takmarkanir og Stigagjöf

- $2 \leq N \leq 500$ .
- Þú mátt í mesta lagi nota  $W \leq 60$  umferðir.

Lausnin þín verður prófuð á safni af prufuhópum og er hver hópur virði einhvers fjölda stiga. Hver prufuhópur inniheldur safn af prufutilvikum. Til að fá stigin fyrir prufuhóp þarftu að leysa sérhvert prufutilvik í prufuhópnum.

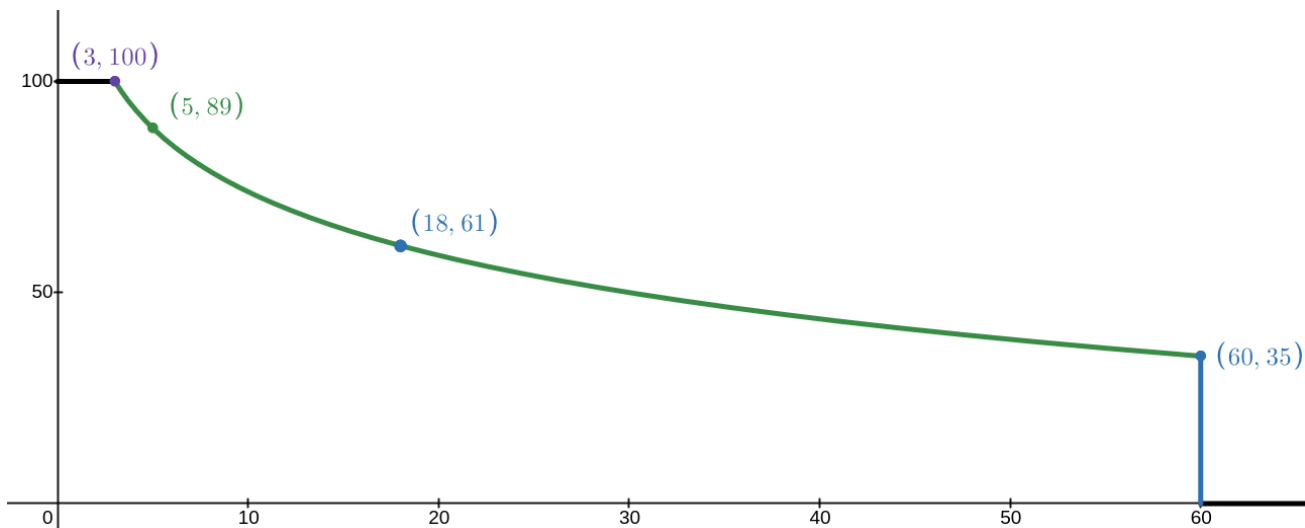
Hópur	Hámarksstig	Takmarkanir
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	Engar frekari takmarkanir

Fyrir hvern prufuhóp sem forritið þitt leysir rétt færðu stig út frá eftirfarandi formúlu:

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

þar sem  $S_g$  er hámarksstig fyrir prufuhópinn og  $W_g$  er hámarksgildið á  $W$  sem var notað í prufutilvikunum í prufuhópnum. Stigin þín fyrir hvern stigahóp verða námunduð að næstu heiltölu.

Grafið að neðan sýnir fjölda stiga, sem fall af  $W$ , sem forritið þitt fær ef það leysir alla prufuhópa með sama gildi af  $W$ . Þá sérstaklega þarftu að leysa sérhvert prufutilvik með  $W \leq 3$  til að fá 100 stig.



## Prufutól

Til að bjóða þér upp á að prófa forritið þitt veitum við þér einfalt tól sem þú getur niðurhalað. Sjáðu viðhengi (“attachments”) neðst á Kattis síðunni. Notkun á tólinu er valkvæm og þú mátt breyta því. Athugaðu að alvöru yfirferðarforritið á Kattis er frábrugðið prufutólinu.

Til að nota tólið skaltu búa til inntaksskrá, til dæmis „sample1.in“, sem skal byrja á heiltölu  $N$  og á eftir fylgir lína með  $N$  heiltölum sem tákna umröðunina og önnur með  $N$  bitum (0 eða 1) sem tákna upprunalegu stöður styttnanna. Til dæmis:

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

Fyrir python forrit, eins og `solution.py` (venjulega keyrð með `pypy3 solution.py`) skal keyra það með:

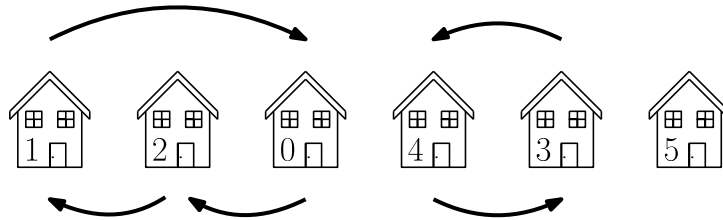
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Fyrir C++ forrit skal fyrst þýða það (t.d. með `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) og svo keyra það með:

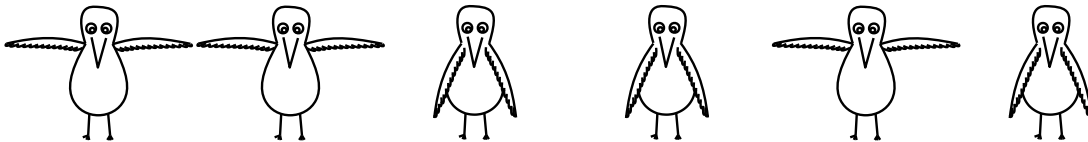
```
python3 testing_tool.py ./solution.out < sample1.in
```

## Sýnidæmi

Í sýnidæminu er eftirfarandi umröðun gefin fyrir íbúa húsanna:

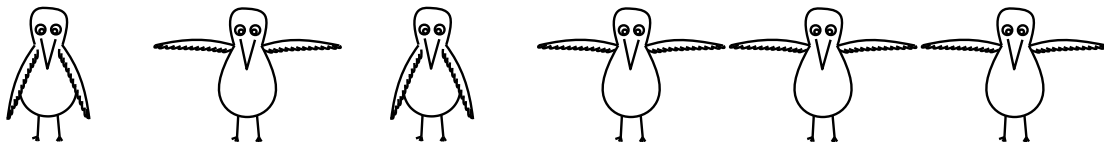


Í fyrsta skiptið er sýnidæmisforritið keyrt (með  $w = 0$ ), og skrifar það út  $W = 2$ , sem þýðir að Arnar mun ganga meðfram götunni tvisvar (og forritið keyrir því tvisvar í viðbót). Fyrir fyrstu gönguna er staða fuglanna eftirfarandi:



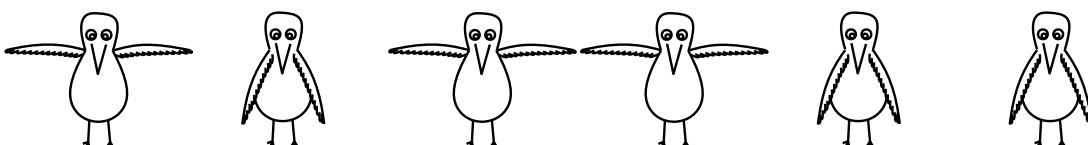
Svo er forritið keyrt með  $w = 1$  fyrir fyrstu göngu Arnars. Hann fer í gegnum fuglanna einn á fætur öðrum og mögulega breytir stöðu þeirra. Sýnidæmisforritið þarf að skrifa út stöðu fugls  $i$  áður en við sjáum fugl  $i + 1$ .

Eftir að Arnar mætir í skólann er staða fuglanna eftirfarandi:



Í síðustu keyrslu forritsins (með  $w = 2$ ), gengur Arnar heim úr skólanum. Mundu að í þessu tilviki fer hann í gegnum fuglanna frá hægri til vinstri eða í öfugri röð. Þetta þýðir að hann þarf að ákvarða stöðu fugls  $i$  áður en hann sér fugl  $i - 1$ .

Eftir að hann klárar gönguna er stað fuglanna eftirfarandi:



Þetta er einmitt rétt stilling. Til dæmis má sjá að fuglastytta 3 (fjórða frá vinstri) er opin ( $b_3 = 1$  núna), sem er rétt því íbúi 4 er að flytja þangað ( $a_3 = 4$ ) og sá íbúi var upprunalega með opna fuglastyttu (upprunalega var  $b_4 = 1$ ).

úttak yfirferðarforritsins	þitt úttak
0 6	
1 2 0 4 3 5	
	2

úttak yfirferðarforritsins	þitt úttak
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

úttak yfirferðarforritsins	þitt úttak
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1