

## D. Баштенски украси

| Име задатка           | Garden Decorations |
|-----------------------|--------------------|
| Временско ограничење  | 7 секунди          |
| Меморијско ограничење | 1 гигабајт         |

Сваког дана када иде у школу и назад кући, Дејте пролази поред улице са  $N$  кућа, нумерисаних од 0 до  $N - 1$ . Тренутно, у кући  $i$  живи особа  $i$ . За промену, становници су одлучили да се међусобно преселе. Особа која ће се уселити у кућу  $i$  је особа  $a_i$  (која тренутно живи у кући  $a_i$ ).

Свака кућа има идентичну статуу птице у врту. Статуе могу бити у два могућа стања: или имају крила отворена (као да птица лети) или затворена (као да стоји на земљи). Становници имају веома јаке преференције о томе како њихове статуе птица треба да изгледају, и одбијају да се преселе у своју нову кућу пре него што статуа у њиховом новом врту изгледа као што је изгледала у њиховом претходном врту. Дејте жели да им помогне да уреде статуе птица како би се могли преселити.

За ово, она ради следеће: кад год шета дуж улице (било на путу до школе или назад кући), посматра птице које пролази и можда подешава неке од статуа (отворивши или затворивши њихова крила). Пошто су њени дани у школи и код куће веома заузети, **не сећа се стања птица које је видела на својим претходним шетњама**. Срећом, она је записала листу  $a_0, a_1, \dots, a_{N-1}$ , тако да зна која се особа сели где.

Помозите Дејте да осмисли стратегију која ће јој рећи које птице да манипулише како би подесила статуе према преференцијама становника. Она може шетати дуж улице највише 60 пута, али да би постигла већи резултат, треба да шета мање пута.

### Имплементација

Ово је проблем са вишеструким извршавањем, што значи да ће ваш програм бити извршен више пута.

У сваком извршавању, треба прво да прочитате ред са два цела броја  $w$  и  $N$ , индекс шетње и број кућа. У првом извршавању вашег програма  $w = 0$ , у другом  $w = 1$ , и тако даље (детаљније објашњење је испод).

На другом реду улаза налази се  $N$  целих бројева  $a_0, a_1, \dots, a_{N-1}$ , што значи да особа која ће се уселити у кућу  $i$  тренутно живи у кући  $a_i$ . Низ  $a_i$  формира пермутацију: то јест, сваки број од 0 до  $N - 1$  појављује се тачно једном у листи  $a_i$ . Напомена, становник може одлучити да се не пресели; то јест,  $a_i = i$  је дозвољено.

Становници се само једном селе. Ово значи да ће за фиксни тест пример, вредност  $N$  и листа  $a_i$  бити исти за сва извршавања вашег програма.

### Прво извршавање.

За прво извршавање вашег програма,  $w = 0$ . У овом извршавању, треба само да испишете један цео број  $W$  ( $0 \leq W \leq 60$ ), број пута који желите да Дејте прошета поред кућа. Ваш програм затим треба да се заврши. Након тога, ваш програм ће бити извршен још  $W$  пута.

### Следећа извршавања.

У следећем извршавању вашег програма,  $w = 1$ ; у наредном  $w = 2$ ; и тако даље до последњег извршавања где је  $w = W$ .

Након што сте прочитали  $w$ ,  $N$  и  $a_0, a_1, \dots, a_{N-1}$ , Дејте почиње да шета дуж улице.

- Ако је  $w$  непаран, Дејте шета од куће до школе и пролази поред кућа редоследом  $0, 1, \dots, N - 1$ .

Ваш програм сада треба да прочита ред са  $b_0$ , било 0 (затворено) или 1 (отворено), тренутно стање статуе испред куће 0. Након што прочитате  $b_0$ , треба да испишете ред са 0 или 1, нову вредност коју желите да поставите  $b_0$ .

Затим ваш програм треба да прочита ред са  $b_1$ , стање статуе испред куће 1; и испише нову вредност  $b_1$ . Ово се наставља за сваку од  $N$  кућа. Након што прођете последњу кућу (тј. прочитате и испишете  $b_{N-1}$ ) ваш програм треба да се заврши.

Напомена, ваш програм може прочитати следећу вредност  $b_{i+1}$  тек након што сте исписали вредност  $b_i$ .

- Ако је  $w$  паран, Дејте шета од школе до куће и пролази поред кућа обрнутим редоследом  $N - 1, N - 2, \dots, 0$ .

Процес је исти као када је  $w$  непаран, осим што почињете са читањем и исписивањем  $b_{N-1}$ , затим  $b_{N-2}$ , и тако даље до  $b_0$ .

Када је  $w = 1$ , улазне вредности  $b_0, b_1, \dots, b_{N-1}$  су оригинално стање статуа птица. Када је  $w > 1$ , улазне вредности  $b_0, b_1, \dots, b_{N-1}$  у ваш програм ће бити оно што је претходно извршавање вашег програма поставило.

На крају, након последњег извршавања вашег програма, вредност  $b_i$  мора бити једнака оригиналној вредности  $b_{a_i}$  за све  $i$ , иначе ћете добити Wrong Answer.

Детаљи.

Ако збир времена извршавања  $W + 1$  одвојених извршавања вашег програма пређе временско ограничење, ваша субмит ће бити оцењена као Time Limit Exceeded.

Обавезно испразните стандардни излаз након штампања сваког реда, иначе ваш програм може бити оцењен као Time Limit Exceeded. У Python-у, ово је обезбеђено аутоматски коришћењем `input()` за читање нових линија. У C++, `cout << endl;` испразни стандардни излаз и испише нову линију; ако користите `printf`, користите `fflush(stdout)`.

## Ограничења и бодовање

- $2 \leq N \leq 500$ .
- Можете користити највише  $W \leq 60$  рунди.

Ваше решење ће бити тестирано на скупу тест група, од којих свака вреди одређен број поена. Свака тест група садржи скуп тест случајева. Да бисте освојили поене за тест групу, потребно је да решите све тест случајеве у тој тест групи.

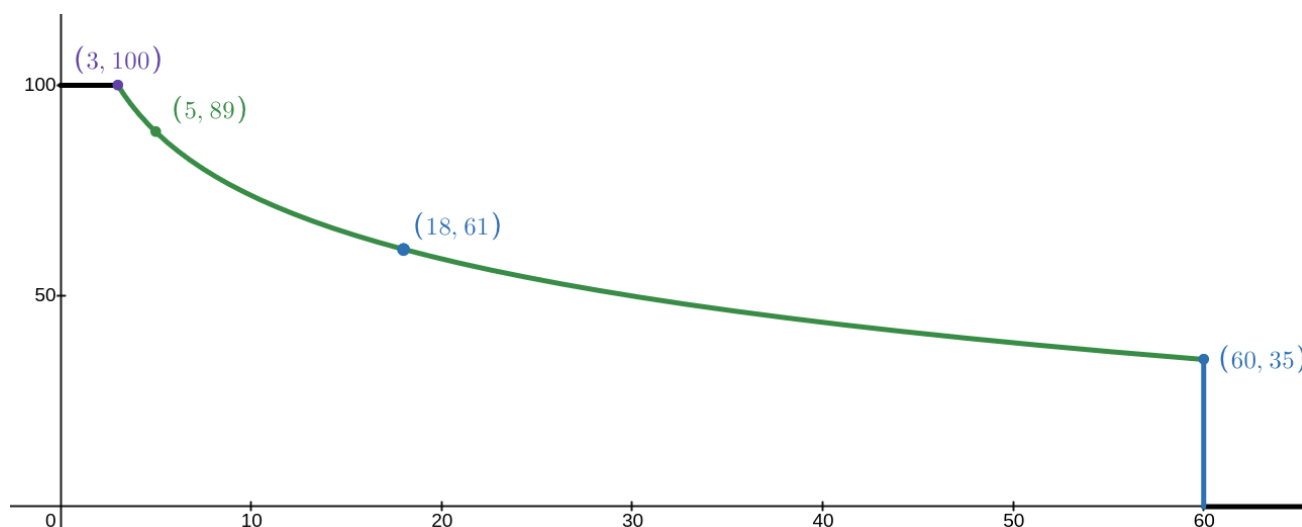
| Групе | Поени | Ограничења               |
|-------|-------|--------------------------|
| 1     | 10    | $N = 2$                  |
| 2     | 24    | $N \leq 15$              |
| 3     | 9     | $a_i = N - 1 - i$        |
| 4     | 13    | $a_i = (i + 1) \bmod N$  |
| 5     | 13    | $a_i = (i - 1) \bmod N$  |
| 6     | 31    | Нема додатних ограничења |

За сваку тест групу коју ваш програм реши исправно, добићете оцену на основу следеће формуле:

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

где је  $S_g$  максимална оцена за тест групу, а  $W_g$  је максимална вредност  $W$  коришћена за било који тест пример у тест групи. Ваша оцена за сваку тест групу биће заокружена на најближи цео број.

Графикон испод показује број поена које ће ваш програм добити ако реши све тест групе са истом вредношћу  $W$ . Посебно, да бисте постигли оцену од 100 поена за овај задатак, потребно је да решите сваки тест случај са  $W \leq 3$ .



## Алат за тестирање

Да би се олакшало тестирање вашег решења, обезбеђујемо једноставан алат који можете преузети. Погледајте "attachments" на дну Kattis странице задатка. Алат је опционо користити. Имајте на уму да је званични програм за оцењивање на Kattis-у различит од алата за тестирање.

Да бисте користили алат, направите улазну датотеку, као што је "sample1.in", која треба да почиње бројем  $N$  након чега следи ред са  $N$  бројева који одређују пермутацију, и још један ред са  $N$  битова (0 или 1) који одређују почетна стања птица. На пример:

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

За Python програме, на пример `solution.py` (који се иначе покрећу са `python3 solution.py`):

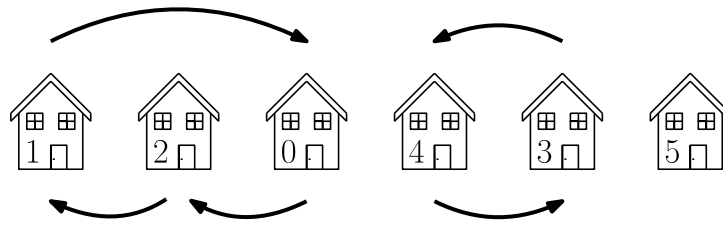
```
python3 testing_tool.py python3 solution.py < sample1.in
```

За C++ програме, прво их компајлирајте (на пример са `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) и онда покрените:

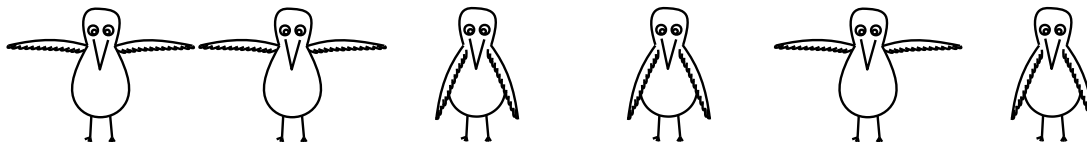
```
python3 testing_tool.py ./solution.out < sample1.in
```

## Примери

У примеру, дате су следеће пермутације људи у кућама:

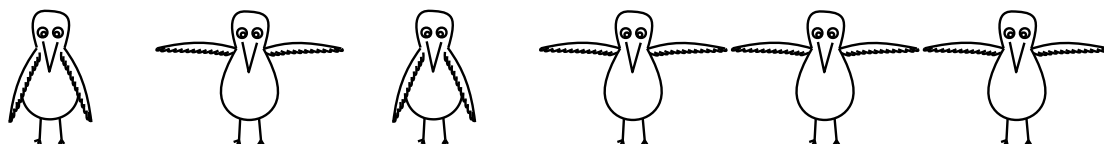


Први пут када се програма извршава (са  $w = 0$ ), он испишује  $W = 2$ , што значи да ће Дејте прошетати улицом два пута (а програм ће се извршити још два пута). Пре прве шетње, птице у вртovima изгледају овако:



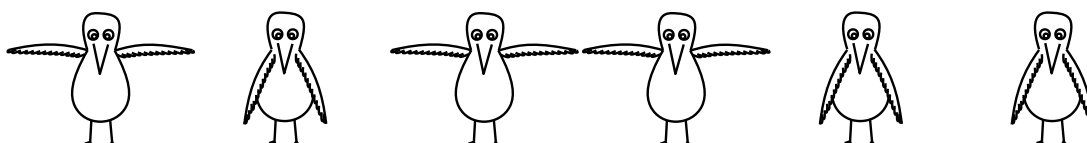
Затим се програм извршава са  $w = 1$ , што указује на прву шетњу Дејте. Она пролази кроз птице једну по једну, почевши с леве стране, и могуће је да промени њихово стање. Програм мора да испише стање  $i$ -те птице пре него што видимо  $(i + 1)$ -ту птицу.

Након што је Дејте стигла у школу, стање птица изгледа овако:



У последњем извршавању програма (са  $w = 2$ ), Дејте се враћа кући из школе. Запамтите да у овом случају, она ће проћи кроз птице с десна на лево и обрадити их у супротном редоследу! Ово значи да треба да одреди стање  $i$ -те птице пре него што види  $(i - 1)$ -ту птицу.

Након што заврши шетњу, птице сада изгледају овако:



Заиста, ово је исправна конфигурација. На пример, статуа птице 3 (тј. четврта слева) је отворена (сада  $b_3 = 1$ ), што је тачно јер ће се особа 4 уселити тамо ( $a_3 = 4$ ) и првобитно је имала отворену статуу птице (првобитно  $b_4 = 1$ ).

| 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           |