# D. Garden Decorations

| Problem Name | Garden Decorations |
|---|---|
| Time Limit | 7 seconds |
| Memory Limit | 1 gigabyte |

Ամեն օր Դատջեն դպրոց գնալու և տուն վերադառնալու ճանապարհին անցնում է $N$ հատ տան մոտով, որոնք համարակալված են $0$-ից $N-1$ թվերով։ Այս պահին, $i$-րդ տան մեջ ապրում է $i$ համարով մարդը։ Տեսարանի փոփոխության համար, բնակիչները որոշել են տներով փոխվել։ Մասնավորապես, $i$-րդ տուն տեղափոխվելու է $a_i$ համարով մարդը (ով հիմա ապրում է $a_i$ համարով տան մեջ)։

Բոլոր տները բակում կա թրչնի արձան։ Արձանների թևերը ունեն երկու հնարավոր վիճակ. *բաց* (կարծես թրչելիս) կամ *փակ* (կարծես գետնի վրա)։ Բնակիչները ցանկանում են, որ իրենց նոր տուն թրչնի արձանը լինի այնպիսի վիճակում ինչպիսին այն իրենց սկզբնական տանն էր, հակառակ դեպքում չեն տեղափոխվի։ Դատջեն ցանկանում է օգնել բնակիչներին և փոխել արձանների վիճակները այնպես, որ բոլորը կարողանան տեղափոխվել։

Դրան հասնելու համար նա անում է հետևյալը. երբ նա անցնում է տների մոտով (դպրոց գնալից կամ տուն վերադառնալիս), նա մեկ առ մեկ տեսնում է տների արձանների վիճակները և կարող է փոխել ընդացիկ տան արձանի վիճակը, եթե ցանկանա (բացելով կամ փակելով թևերը)։ Քանի, որ նա շատ զբաղված է լինում տանը և դպրոցում, **նա չի հիշում արձանների վիճակները իր նախորդ անցումների ժամանակ**։ Բարեբախտաբար, նա իր տետրում գրել է $a_0, a_1, ..., a_{N-1}$ հաջորդականությունը, այսինքն գիտի թե, որ բնակիչը, որ տուն է տեղափոխվելու։

Օգնեք Դետջային այնպիսի սրատեգյա մտածել, որ հնարավոր լինի բոլոր բնակիչների պահանջները բավարարել որոշ անցումներից հետո։ Նա կարող է անցնել տների մոտով առավելագույնը $60$ անգամ, բայց ավելի բարձր միավոր ստանալու համար նա ավելի քիչ անգամ պետք է անցնի ճանապարհը։

## Իրականացում

Սա multirun խնդիր է, այսինքն մեկ թեստը ստուգելու համար Ձեր ծրագիրը մի քանի անգամ է աշխատացվելու։

Երբ Ձեր ծրագիրը աշխատացվում է, սկզբում պետք է կարդաք երկու թիվ՝ $w$ և $N$. անցման ինդեքսը և տների քանակը: Ձեր ծրագրի առաջին անգամ աշխատացնելու ժամանակ $w = 0$, երկրորդ անգամ աշխատացվելու ժամանակ $w = 1$, և այդպես շարունակ (մանրամասնությունները տրված են ստորև):

Երկրորդ տողում տրված են $N$ հատ թվեր՝ $a_0, a_1, ..., a_{N-1}$. $i$ համարի տուն պետք է տեղափոխվի $a_i$ համարով մարդը: $a_i$-երը կազմում են *տեղափոխություն*. այսինքն 0-ից $N-1$ թվերը հանդիպում են ճիշտ մեկ անգամ: Նկատեք, որ բնակիչը կարող է չտեղափոխվել, այդ դեպքում $a_i = i$:

Մեկ թեստի ժամանակ բնակիչները կատարվում է բնակիչների մեկ տեղափոխություն: Սա նշանակում է, որ $N$-ի և $a_i$-երի արժեքները մեկ թեստի ստուգման ժամանակ Ձեր ծրագիրը աշխատացնելիս չեն փոխվում:

## Առաջին անգամ աշխատացնելիս...

Առաջին անգամ աշխատացնելիս $w = 0$: Այս դեպքում ուղղակի պետք է տպել մեկ թիվ՝ $W$ ($0 \leq W \leq 60$), Դետջեի անցումների քանակը (քանի անգամ Ձեր ծրագիրը պետք է աշխատացվի): Այնուհետև Ձեր ծրագիրը պետք է ավարտվի: Սրանից հետո Ձեր ծրագիրը աշխատացվելու է ևս $W$ անգամ:

## Հաջորդ անգամներում...

Հաջորդ աշխատացնելու ժամանակ $w = 1$, երրորդ աշխատացնելու ժամանակ $w = 2$, և այդպես շարունակ մինչև վերջին անգամը, որտեղ $w = W$:

$w$, $N$ և $a_0, a_1, ..., a_{N-1}$ թվերը կարդալուց հետո, Դետջեն սկսում է քայլել ճանապարհի երկայնքով:

- Եթե $w$-ն կենտ է, Դետջեն գնում է դպրոց և տներով անցնելու է $0, 1, ..., N-1$ հերթականությամբ:

  Հիմա Ձեր ծրագիրը պետք է կարդա $b_0$ թիվը, որը կամ $0$ է (փակ թևերով) կամ $1$ է (բաց թևերով). $0$ համարով տան առաջ գտնվող արձանի վիճակը: $b_0$ թիվը կարդալուց հետո, Դուք պետք է տպեք մեկ տող որը պարունակում է $0$ կամ $1$. $b_0$-ի նոր արժեքը:

  Այնուհետև Ձեր ժրագիրը պետք է կարդա $b_1$ թիվը. $1$ համարով տան արձանի վիճակը; և տպի $b_1$-ի նոր արժեքը: Սա արվում է $N$ տներից յուրաքանչյուրի համար: Վերջին տան մոտով անցնելուց հետո (այսինքն $b_{N-1}$-ը կարդալուց և դրա նոր արժեքը որոշելուց հետո) **Ձեր ծրագիրը պետք է ավարտվի**:

  *Նկատեք, որ Ձեր ծրագիրը կարող է կարդալ $b_{i+1}$ թիվը միայն $b_i$ թվի նոր արժեքը որոշելուց հետո:*

- Եթե $w$ թիվը զույգ է, Դետջեն գալիս է տուն և տներով անցնելու է $N-1, N-2, ...0$ հերթականությամբ:

  Պրոցեսը համընկնում է $w$-ի կենտ լինելու պրոցեսի հետ, միայն սկզբում մշակվելու է $b_{N-1}$-ը, հետո $b_{N-2}$-ը, և այդպես շարունակ, իսկ վերջում $b_0$-ն:

Երբ $w = 1$, $b_0, b_1, ..., b_{N-1}$ հաջորդականությունը ներկայացնում է արձանների սկզբմանական վիճակները: Երբ $w > 1$, Ձեզ տրվող $b_0, b_1, ..., b_{N-1}$ թվերը համընկնում են Ձեր ծրագրի նախորդ աշխատանքից հետո ստացված թվերի հետ:

Ձեր ծրագրի վերջին աշխատանքից հետո $b_i$-ն պետք է հավասար լինի սկզբնական $b_{a_i}$-ին բոլոր $i$-ի համար, հակառակ դեպքում կստանաք Wrong Answer:

### Մանրամասներ

Եթե Ձեր ծրագրի $W+1$ աշխատանքների ժամանկների *գումարը* գերազանցում է ժամանակի սահմանափակումը, ապա կստանաք Time Limit Exceeded:

Հիշեք, որ կամայական տող տպելուց հետո պետք է մաքրել ստանդարտ ելքի բուֆերը, հակառակ դեքում կարող եք ստանալ Time Limit Exceeded: Python լեզվում, սա տեղի է ունենում ավտոմատ: C++ լեզվում, `cout << endl;`-ը մաքրում է ստանդարտ ելքի բուֆերը նոր տողի անցնելու հետ մեկտեղ, իսկ printf օգտագործելու դեպքում, օգտագործեք `fflush(stdout)`:

## Սահմանափակումներ և Գնահատում

- $2 \leq N \leq 500$.
- Կարող եք օգտագործել առավելագույնը $W \leq 60$ աշխատեցումներ:

Ձեր լուծումը թեստավորվելու է մի քանի թեստերի խմբերի վրա (ենթախնդիր), որոնցից յուրաքանչյուրը կարող է տալ որոշակի միավոր: Ամեն խումբ պարունակում է որոշակի քանակությամբ թեստեր: Ենթախնդրից միավոր ստանալու համար պետք է այդ ենթախնդրի բոլոր թեստերը լուծված լինեն:

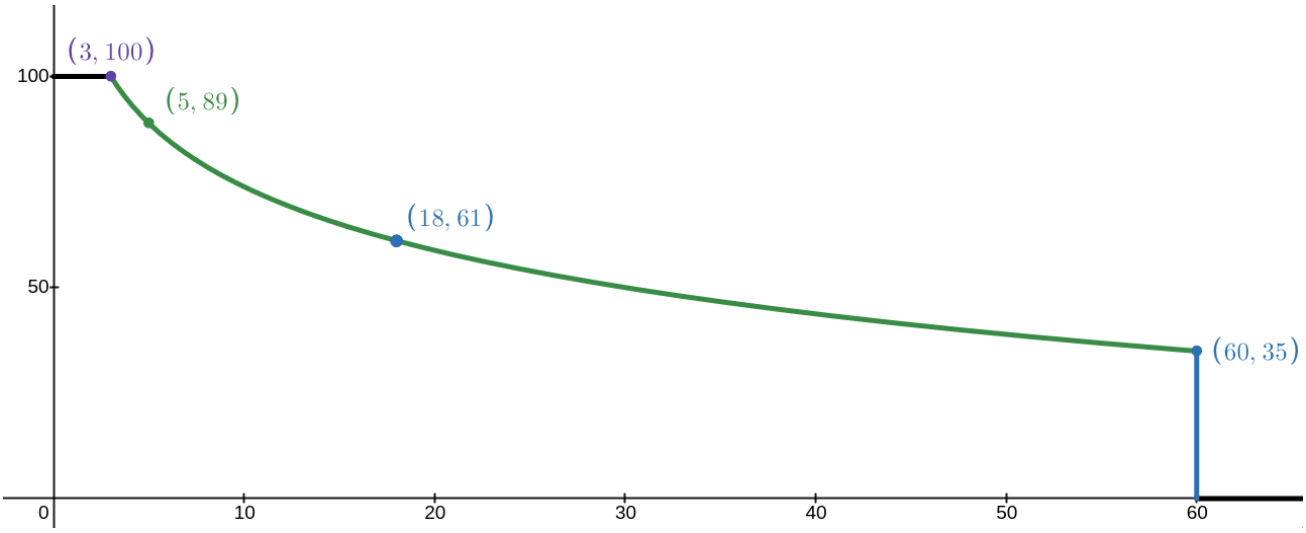
| Խումբ | Առավելագույն միավոր | Սահմանափակումներ |
|---|---|---|
| 1 | 10 | $N = 2$ |
| 2 | 24 | $N \leq 15$ |
| 3 | 9 | $a_i = N - 1 - i$ |
| 4 | 13 | $a_i = (i + 1) \bmod N$ |
| 5 | 13 | $a_i = (i - 1) \bmod N$ |
| 6 | 31 | Լրացուցիչ սահմանափակումներ չկան |

Եթե Ձեր լուծումը ճիշտ է աշխատել ենթախնդրի բոլոր թեստերի վրա, ապա կստանա հետևյալ միավորը.

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

որտեղ $S_g$-ն տվյալ ենթախնդրի առավելագույն միավորն է, և $W_g$-ն Ձեր գոտագործած $W$-ի առավելագույն արժեքը, այս ենթախնդրի թեստերի վրա: Միավորը կկլորացվի միչև մոտակա ամբողջ թիվը:

Հետևյան ֆունկցիաի գրաֆիը նկարագրում է միավորների քանակը, որը Ձեր ծրագիրը կստանար, եթե բոլոր թեստերը լուծեր $W$ աշխատացում օգտագործելով: Մասնավորապես 100 միավոր ստանալու համար պետք է օգտագործել $W \leq 3$ աշխատեցում:




## Թեստավորման գործիք

Լոկալ ստուգումը հեշտացնելու համար Ձեզ տրվում է գործիք: Գործիքը կարող եք ներբեռնել “attachments” բաժնում որը գտվում է Kattis problem page-ի ներքևում: Գործիքի օգտագործումը պարտադիր չէ: Ձեր ծրագիրը ստուգվելու է մեկ այլ գործիքի միջոցով:

Գործիքը օգտագործելու համար ստեղծեք input file, օրինակ՝ “sample1.in”, որի առաջին տողում պետք է գրված լինի $N$ թիվը, որին հաջորդում է $N$ հատ թվեր, որոնք իրենցից ներկայացնում են տեղափոխություն ($a$ հաջորդականությունը), ապա պետք է տրված լինեն $N$ հատ բիթեր ($0$ կամ $1$). արձանների սկզբնական վիճակները: Օրինակ.

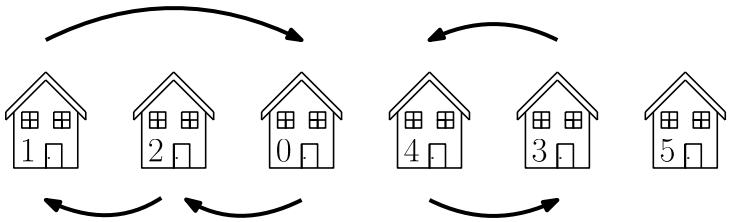
```
6
1 2 0 4 3 5
1 1 0 0 1 0
```

C++ Լեզվի համար, սկզբում կոմպիլացրեք ծրագիրը (օրինակ օգտագործելով `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out` հրամանը), այնուհետև գործիքը օգտագործեք հետևյալ հրամանով.

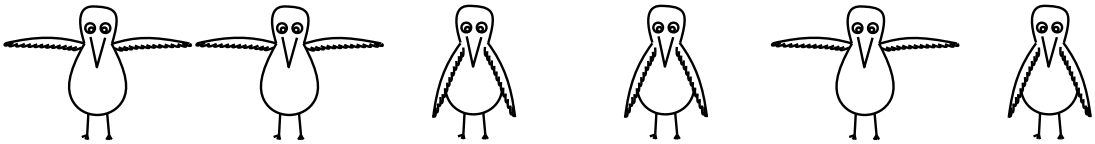
```
python3 testing_tool.py ./solution.out < sample1.in
```

## Օրինակ

Օրինակում տրված է մարդկանց հետևյալ տեղափոխությունը.

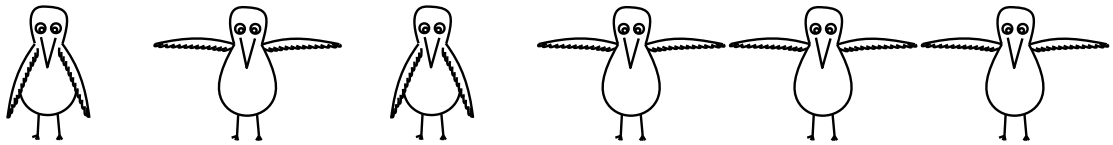



Առաջին անգամ աշխատացնելիս ($w = 0$), ծրագիրը տպում է $W = 2$, այսինքն Դետջեն անցնելու է երկու անգամ (և ծրագիրը աշխատացվելու է ևս երկու անգամ)։ Առաջին անցումից առաջ արձանները ունեն հետևյալ վիճակները.



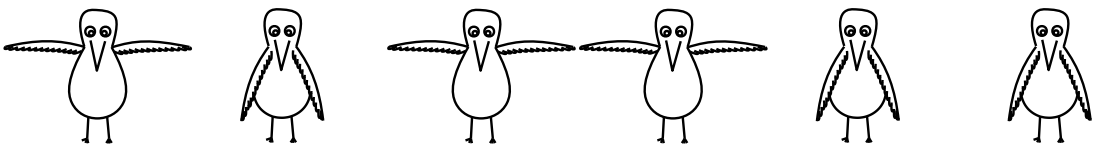
Այնուհետև ծրագիրը աշխատացվում է $w = 1$-ով. ազդարարելով, որ սա Դետջեի առաջին անցումն է։ Նա անցնում է արձանների կողքով մեկ առ մեկ, ձախից աջ, և փոփոխում է որոշ արձանների վիճակները։ Ծրագիրը պետք է տպի $i$-րդ արձանի վիճակը $(i + 1)$-իը վիճակը կարդալուց առաջ։

Դետջեի դպրոց հասնելուց հետո վիճակները այսպիսին են.



Վերջին կանչի ժամանակ ($w = 2$) Դետջեն վերադառնում է դպրոցից տուն։ Այս դեպքում նա անցնում է աջից ձախ։ Այսինքն նա պետք է որոշի $i$-րդ արձանի վիճակը $(i - 1)$-ը տեսնելուց առաջ։

Տուն վերադառնալուց հետո արձանները ունեն հետևյալ տեսքը.



Իրոք, սա ճիշտ վիճակներ են։ օրինակ $b_3 = 1$, ինչը ճիշտ է, որովհետև $4$ համարի մարդ պետք է տեղափոխվի այդտեղ ($a_3 = 4$) և նա ի սկզբանե ուներ բաց ձեռքերով արձան (իսկզբանե $b_4 = 1$).

| 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 |