D. ბაღის დეკორაციები

ამოცანის სახელი	ბაღის დეკორაციები
დროის ლიმიტი	7 წამი
მეხსიერების ლიმიტი	1 გბ

დეტიე ყოველდღე სკოლაში წასვლისას და სახლში დაბრუნებისას გადის ქუჩას, რომელზეც 0-დან $(N-1)$-მდე გადანომრილი N რაოდენობის სახლია ჩამწკრივებული. ამჟამად i-ურ სახლში i-ური მაცხოვრებელი ცხოვრობს. გარემოების გამოსაცვლელად ქუჩაზე მაცხოვრებლებმა სახლების ერთმანეთში გაცვლა გადაწყვიტეს, სადაც i-ურ სახლში a_i-ური ადამიანი გადავა საცხოვრებლად (რომელიც ახლა a_i-ურ სახლში ცხოვრობს).

თითოეულის სახლის ბაღში არის ჩიტის ქანდაკება.

ქანდაკებებს აქვთ ორი შესაძლო მდგომარეობა: მათი ფრთები ან *გაშლილია* (ჩიტი დაფრინავს) ან *დაშვებული* (ჩიტი მიწაზე დგას). მაცხოვრებლებს აქვთ დიდი მოთხოვნილებები იმაზე, თუ როგორ უნდა გამოიყურებოდნენ მათი ქანდაკებები. ამჟამად i-ური სახლის ეზოს ქანდაკება i-ური მაცხოვრებლის სასურველ მდგომარეობაშია.

მაცხოვრებლები უარს ამბობენ ისეთ სახლში გადასვლაზე, სადაც ქანდაკება მათთვის არასასურველ მდგომარეობაშია. დეტიეს უნდა, რომ დაეხმაროს მათ ქანდაკებების მდგომარეობების შეცვლაში ისე, რომ მათ სახლების გაცვლა შეეძლოთ.

ამისათვის, ის აკეთებს შემდეგს: როდესაც ჩაივლის ქუჩას (სკოლისკენ ან უკან სახლისკენ), ის აკვირდება ჩიტის ქანდაკებებს სათითაოდ და სურვილისამებრ ცვლის მათ მდგომარეობას (ფრთების გახსნით ან დაშვებით).

რადგან ის სკოლაში და სახლში ძალიან დატვირთულია, **მას არ ახსოვს წინა გავლებისას დანახული ქანდაკებების მდგომარეობები.** საბედნიეროდ, დეტიეს ჩამოწერილი აქვს სია $a_0, a_1, ..., a_{N-1}$ და შესაბამისად იცის რომელი მაცხოვრებელი რომელ სახლში გადადის.

დაეხმარეთ დეტიეს გამოიმუშავოს სტრატეგია, რომელიც უკარნახებს მას, თუ რომელი ქანდაკებების მდგომარეობები შეცვალოს, რომ დააკმაყოფილოს მაცხოვრებლების მოთხოვნილებები. მას შეუძლია ქუჩის გავლა მაქსიმუმ 60-ჯერ, მაგრამ მაღალი ქულის ასაღებად, მან უნდა გაიაროს ქუჩა რაც შეიძლება ნაკლებჯერ.

იმპლემენტაცია

ეს არის მრავლგაშვებადი ამოცანა, რაც იმას ნიშნავს რომ თქვენი პროგრამა გაშვებული იქნება რამდენჯერმე.

თითოეულ გაშვებაში უნდა წაიკითხოთ სტრიქონი ორი მთელი რიცხვით: w და N, ქუჩის გავლის ინდექსი და სახლების რაოდენობა. პროგრამის პირველ გაშვებაში $w = 0$, მეორეში $w = 1$ და ასე შემდეგ (დეტალები იხილეთ ქვემოთ).

მეორე ხაზში მოცემულია N რაოდენობის $a_0, a_1, ..., a_{N-1}$ მთელი რიცხვი, სადაც მაცხოვრებელი, რომელიც გადადის i-ურ სახლში, ამჟამად ცხოვრობს a_i-ურ სახლში. a_i-ები ქმნიან *პერმუტაციას*, რაც ნიშნავს, რომ თითოეული რიცხვი 0-დან $(N-1)$-ის ჩათვლით გვხვდება ზუსტად ერთხელ a_i-ების მასივში.

ყურადღება მიაქციეთ, რომ შესაძლოა მაცხოვრებელმა იგივე სახლში დარჩენა გადაწყვიტოს. რაც ნიშნავს, რომ $a_i = i$ დასაშვებია.

მაცხოვრებლები სახლებს მხოლოდ ერთხელ იცვლიან, რაც ნიშნავს, რომ კონკრეტული ტესტისთვის N-ის მნიშვნელობა და a_i-ების მასივი იქნება შეუცვლელი თქვენი პროგრამის ყველა გაშვებისათვის.

პირველი გაშვება

თქვენი პროგრამის პირველი გაშვებისას $w = 0$. ამ გაშვებაში თქვენ უბრალოდ უნდა გამოიტანოთ მთელი რიცხვი W ($0 \leq W \leq 60$) - რაოდენობა რამდენჯერაც გსურთ, რომ დეტიემ გაიაროს ქუჩა. შემდეგ თქვენი პროგრამა უნდა დასრულდეს. ამის შემდეგ თქვენი პროგრამა იქნება თავიდან გაშვებული W-ჯერ.

შემდეგი გაშვებები

თქვენი პროგრამის შემდეგ გაშვებაში $w = 1$, მომდევნოში $w = 2$ და ასე შემდეგ ბოლო გაშვებამდე, სადაც $w = W$.

როდესაც წაიკითხავთ w, N და $a_0, a_1, ..., a_{N-1}$, დეტიე იწყებს სიარულს ქუჩის გასწვრივ.

- თუ w კენტია, დეტიე გადის ქუჩას სახლიდან სკოლამდე და გაივლის სახლებს თანმიმდევრობით $0, 1, ..., N-1$.

  თქვენმა პროგრამმა ახლა უნდა წაიკითხოს სტრიქონი b_0-ით, რომელიც უდრის 0-ს (დახურული) ან 1-ს (დაშვებული), რაც აღნიშნავს სახლი 0-ის ქანდაკების ახლანდელ მდგომარეობას. b_0-ს წაკითხვის შემდეგ, უნდა გამოიტანოთ ახალი სტრიქონი რომელიც შეიცავს 0-ს ან 1-ს, ახალი მდგომარეობა რომელიც გინდათ რომ მიანიჭოთ b_0-ს.

  შემდეგ თქვენმა პროგრამამ უნდა წაიკითხოს სტრიქონი b_1-ით, სახლი 1-ის ქანდაკების მდგომარეობა და გამოიტანოს b_1-ის ახალი მნიშვნელობა.

ეს გრძელდება ყოველი შემდეგი N სახლისთვის. როცა დეტიე ბოლო სახლს გაივლის (ანუ, თქვენ წაიკითხავთ და გადავწერთ b_{N-1}-ს) **თქვენი პროგრამა უნდა დამთავრდეს**.

ყურადღება მიაქციეთ, რომ თქვენს პროგრამას შეუძლია წაიკითხოს შემდეგი b_{i+1}-ის მნიშვნელობა მხოლოდ b_i-ის მნიშვნელობის გამოტანის შემდეგ.

- თუ w ლუწია, დეტიე გადის ქუჩას სკოლიდან სახლამდე და გაივლის სახლებს შებრუნებული თანმიმდევრობით $N-1, N-2, ...0$. პროცესი იგივეა როდესაც w არის კენტი, უბრალოდ იწყებთ კითხვას და გადაწერას b_{N-1} მნიშვნელობის, შემდეგ b_{N-2}-ის, და ასე შემდეგ b_0-მდე.

როდესაც $w = 1$, შესატანი მნიშვნელობები $b_0, b_1, ..., b_{N-1}$ არის ქანდაკებების საწყისი მდგომარეობები (რომელიც ასევე შეესაბამება შესაბამისი მაცხოვრებლების ქანდაკებების სასურველ მდგომარეობებს). როდესაც $w > 1$, შეტანილი მნიშვნელობები $b_0, b_1, ..., b_{N-1}$ თქვენს პროგრამაში იქნება წინა გაშვების დროს გამოტანილი მნიშვნელობების შესაბამისი .

საბოლოოდ, თქვენი პროგრამის საბოლოო გაშვების შემდეგ, b_i-ის მნიშვნელობა უნდა უდრიდეს b_{a_i}-ის საწყის მნიშვნელობას ყველა i-სათვის, სხვა შემთხვევაში ამოხსნა შეფასდება როგორც Wrong Answer.

დეტალები

თუ თქვენი პროგრამის $W + 1$ ცალი დამოუკიდებელი გაშვების მიერ გამოყენებული დროების ჯამი აღემატება ამოცანის დროის ლიმიტს, თქვენი ამოხსნა იქნება შეფასებული როგორც Time Limit Exceeded.

Make sure to flush standard output after printing each line, or else your program might get judged as Time Limit Exceeded. In Python, this happens automatically as long as you use `input()` to read lines. In C++, `cout << endl;` flushes in addition to printing a newline; if using printf, use `fflush(stdout)`.

შეზღუდვები და შეფასება

- $2 \leq N \leq 500$.
- შეგიძლიათ გამოიყენოთ მაქსიმუმ $W \leq 60$ რაუნდი.

თქვენი ამოხსნა შემოწმდება ტესტების ჯგუფთა ნაკრებზე, რომელთაგან თითოეულზე ქულათა გარკვეულ რაოდენობას მიიღებთ. ტესტების ყოველი ჯგუფი შეიცავს განსაზღვრული რაოდენობის ტესტებს და თითოეულ ჯგუფზე ქულების მისაღებად აუცილებელია თქვენი ამოხსნა ამ ჯგუფში შემავალ ყველა ტესტზე სწორ პასუხს იძლეოდეს.

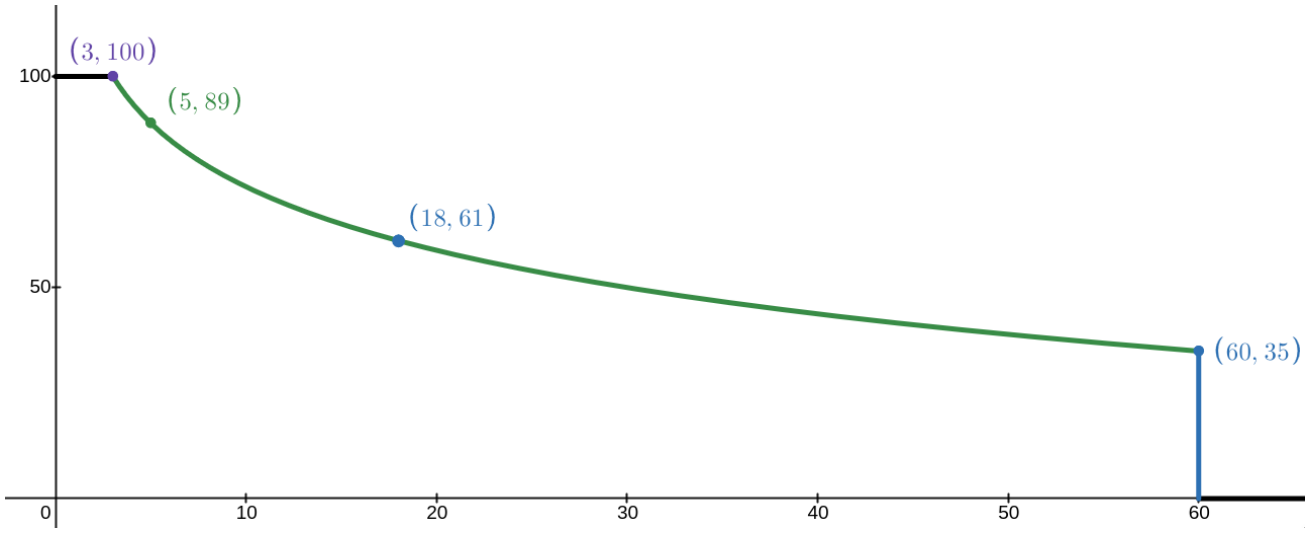
ჯგუფი	მაქსიმალური ქულა	შეზღუდვები
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	დამატებითი შეზღვუდვების გარეშე

ყველა ტესტის ჯგუფისთვის, რომელსაც თქვენი პროგრამა სწორად ამოხსნის თქვენ მიიღებთ ქულას შემდეგი ფორმულის მიხედვით:

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

სადაც S_g არის მაქსიმალური ქულა ტესტის ჯგუფისთვის და W_g არის გამოყენებული W-ების მაქსიმალური მნიშვნელობა ამ ტესტის ჯგუფის ტესტებს შორის. თქვენი თითოეული ტესტის ჯგუფის ქულა დამრგვალებული იქნება უახლოს მთელ რიცხვამდე.

ქვევით ნახაზზე გამოსახულია ქულების რაოდენობა, როგორც W-ზე დამოკიდებული ფუნქცია, რასაც თქვენი პროგრამა აიღებს, თუ ყველა ტესტის ჯგუფს ამოხსნის იგივე W-თი. უფრო კონკრეტულად, ამ ამოცანაზე 100 ქულის მისაღებად, თქვენ უნდა ამოხსნათ ყველა ტესტი $W \leq 3$ რაუნდების რაოდენობით.



ტესტირების საშუალება

რომ დაგეხმაროთ თქვენი ამოხსნის გატესტვაში, გთავაზობთ მარტივ საშუალებას, რომლის გადმოწერაც შეგიძლიათ. იხილეთ “attachments” Kattis-ზე ამოცანის გვერდის ბოლოში. ამ

ტესტირების საშუალების გამოყენება სავალდებულო არაა. გაითვალისწინეთ, რომ ოფიციალური გრადერი პროგრამა Kattis-ზე განსხვავდება ამ ტესტირების საშუალებისგან.

ამ საშუალების გამოსაყენებლად, შექმენით input ფაილი. მაგალითად, “sample1.in”, რომელიც უნდა დაიწყოთ N-ით და რომელსაც მოყვება N ცალი რიცხვი. ეს N ცალი რიცხვი ასახავს პერმუტაციას და შემდეგი სტრიქონი შედგება N ცალი ბიტისაგან (0 ან 1), რომლებიც ასახავენ ქანდაკებების საწყის მდგომარეობებს. მაგალითად:

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

პითონის პროგრამებისთვის, მაგალითად, `solution.py` (ტერმინალში ამოხსნას ვუშვებთ როგორც `pypy3 solution.py`):

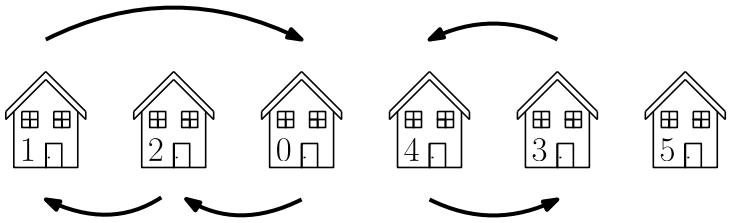
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++-ის პროგრამებისთვის და-compile-ების შემდეგ (და-compile-ება შეიძლება ტერმინალში შემდეგი სკრიპტის გაშვებით: `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) გაუშვით (ასევე ტერმინალში):

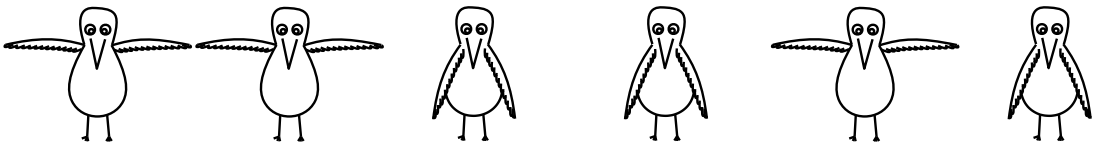
```
python3 testing_tool.py ./solution.out < sample1.in
```

მაგალითი

ამ მაგალითში, ჩვენ გვაძლევენ ხალხის შემდეგ პერმუტაციას სახლებში:



პროგრამის პირველ გაშვებაზე (როდესაც $w = 0$), მას გამოაქვს $W = 2$, რაც ნიშნავს რომ დეტიე გაივლის ქუჩას ორჯერ (და პროგრამა გაშვებული იქნება კიდევ ორჯერ). ქუჩის პირველ გავლამდე, ჩიტის ქანდაკებები ბაღებში გამოიყურება შესაბამისად:

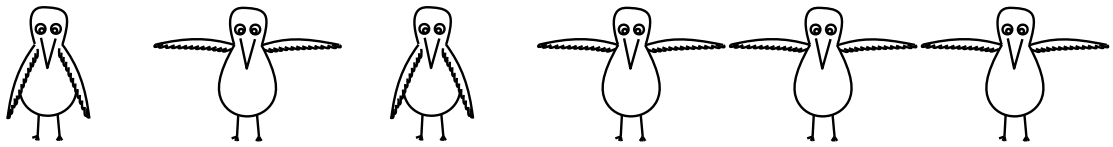


შემდეგ პროგრამა გაშვებულია როგორც $w = 1$, რაც აღნიშნავს დეტიეს მიერ ქუჩის პირველად გავლას. ის იწყებს მარცხნივ, გადის ქანდაკებებს სათითაოდ და პოტენციურად ცვლის მათ

მდგომარეობებს.

პროგრამამ უნდა გამოიტანოს i-ური ჩიტის ქანდაკების მდგომარეობა მანამდე, სანამ შეამოწმებთ $(i+1)$-ე ჩიტს.

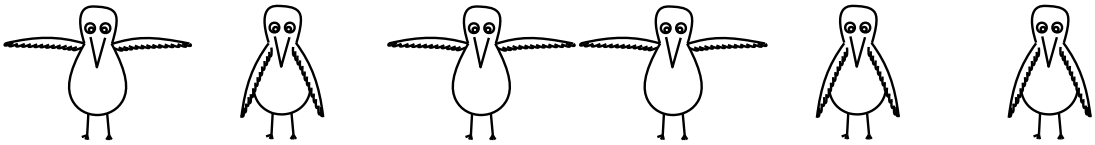
როდესაც დეტიე სკოლაში მივა, ქანდაკებების მდგომარეობა შემდეგია:



პროგრამის საბოლოო გაშვების დროს (როცა $w = 2$), დეტიე სკოლიდან სახლში ბრუნდება. ამ შემთხვევაში ის გაივლის ქანდაკებებს მარჯვნიდან მარცხნივ და დაამუშავებს მას შესაბამისად საპირისპირო მიმდევრობით.

პროგრამამ უნდა გამოიტანოს i-ური ჩიტის ქანდაკების მდგომარეობა მანამდე, სანამ შეამოწმებთ $(i-1)$-ე ჩიტს.

როდესაც ის დაასრულებს ქუჩის გავლას, ქანდაკებების მდგომარეობა შემდეგია:



ნამდვილად, ეს არის სწორი განლაგება. მაგალითად, ჩიტის ქანდაკება 3 (ანუ მარცხნიდან მეოთხე) არის გაშლილი ფრთებით (ახლა $b_3 = 1$), რაც სწორია, რადგან მოსახლე 4 გადავა ამ სახლში საცხოვრებლად ($a_3 = 4$) და მას დასაწყისში ჰქონდა გაშლილ ფრთებიანი ჩიტის ქანდაკება (დასაწყისში $b_4 = 1$).

გრადერის მიერ გამოტანილი პასუხი	თქვენი გამოტანილი პასუხი
0 6	
1 2 0 4 3 5	
	2

გრადერის მიერ გამოტანილი პასუხი	თქვენი გამოტანილი პასუხი
1 6	
1 2 0 4 3 5	
1	
	0
1	
	1
0	
	0
0	
	1
1	
	1
0	
	1

გრადერის მიერ გამოტანილი პასუხი	თქვენი გამოტანილი პასუხი
2 6	
1 2 0 4 3 5	
1	
	0
1	
	0
1	
	1
0	
	1
1	
	0
0	
	1