D. Dekorasi Taman

Judul Soal	Dekorasi Taman
Batas Waktu	7 detik
Batas Memori	1 gigabyte

Setiap harinya ketika ia pergi dan pulang dari sekolah, Detje berjalan sepanjang sebuah jalan dengan N rumah, yang dinomori dari 0 hingga $N-1$. Pada saat ini, rumah i sedang ditinggali oleh warga i. Untuk ganti suasana, para warga pun memutuskan untuk bertukaran rumah dengan yang lainnya. Warga yang akan pindah ke rumah i adalah warga a_i (yang sekarang tinggal di rumah a_i).

Setiap rumah memiliki sebuah patung burung yang identik di taman mereka. Patung-patung tersebut memiliki dua kemungkinan status, burung dengan sayap yang terbuka (seperti sedang terbang) atau dengan sayap yang tertutup (seperti sedang berdiri di tanah). Para warga memiliki preferensi yang sangat kuat terhadap bagaimana bentuk patung burung mereka. Pada saat ini, patung burung yang terdapat di depan rumah i memiliki status yang sesuai preferensi warga i. Para warga akan menolak untuk pindah ke rumah baru kecuali patung di taman yang baru sudah berbentuk sesuai dengan preferensi mereka. Detje ingin membantu mereka menata patung-patung burung tersebut agar mereka bisa pindah.

Untuk ini, ia melakukan hal berikut: ketika ia berjalan sepanjang jalan (antara di perjalanannya ke sekolah atau pulang dari sekolah), ia mengamati patung-patung burung yang ia lewati dan bisa jadi menyesuaikan status beberapa patungnya (dengan membuka atau menutup sayap). Karena hari-harinya di sekolah dan di rumah sangatlah padat, **ia tidak mengingat status dari patung burung yang ia lihat di perjalananan sebelum-sebelumnya**. Untungnya, ia telah menuliskan sebuah daftar $a_0, a_1, ..., a_{N-1}$ agar ia mengetahui warga yang mana pindah ke rumah mana.

Bantulah Detje merancang sebuah strategi yang memberitahunya patung burung mana yang perlu dimanipulasi untuk menyesuaikan sesuai preferensi warga. Dia bisa berjalan sepanjang jalan tersebut paling banyak 60 kali, namun untuk mendapatkan nilai yang lebih tinggi, ia perlu berjalan sepanjang jalan tersebut lebih sedikit.

Implementasi

Soal ini merupakan soal *multirun*, yang berarti program Anda akan dieksekusi beberapa kali.

Untuk setiap eksekusi, program Anda perlu membaca sebuah baris dengan dua buah bilangan bulat w dan N, indeks perjalanan dan banyaknya rumah. Pada eksekusi pertama program Anda, $w = 0$. Pada eksekusi kedua, $w = 1$, dan seterusnya (detail lebih lanjut akan dijelaskan di bawah).

Pada baris kedua masukan, terdapat N buah bilangan bulat $a_0, a_1, ..., a_{N-1}$, yang berarti warga yang akan pindah ke rumah i sedang tinggal di rumah a_i. a_i merupakan sebuah *permutasi*: artinya, setiap bilangan dari 0 hingga $N - 1$ akan muncul tepat sekali di a_i. Catat bahwa warga bisa memilih untuk tidak pindah; Artinya, $a_i = i$ diperbolehkan.

Para warga hanya akan pindah rumah sekali. Hal ini berarti untuk kasus uji yang tetap, nilai dari N dan daftar a_is akan selalu sama untuk semua eksekusi program Anda.

Proses Pertama.

Pada eksekusi pertama program Anda, $w = 0$.

Pada eksekusi ini, Anda hanya perlu mengeluarkan sebuah bilangan bulat W ($0 \leq W \leq 60$), banyaknya perjalanan melewati rumah-rumah yang perlu dilakukan oleh Detje.

Kemudian, program Anda perlu berhenti. Mengikuti ini, program Anda akan dijalankan kembali W kali lagi.

Proses Setelahnya.

Pada eksekusi program Anda selanjutnya, $w = 1$; pada eksekusi berikutnya $w = 2$; dan seterusnya hingga eksekusi terakhir dengan $w = W$.

Setelah Anda telah membaca w, N, dan $a_0, a_1, ..., a_{N-1}$, Detje akan mulai berjalan sepanjang jalan.

- Jika w adalah ganjil, Detje akan berjalan dari rumah ke sekolahnya, dan dia akan melewati rumah-rumah dengan urutan $0, 1, ..., N - 1$. Program Anda perlu membaca sebuah baris dengan b_0, antara 0 (tertutup) atau 1 (terbuka), status sekarang dari patung yang ada di depan rumah 0. Setelah Anda membaca b_0, Anda perlu mengeluarkan sebuah baris dengan 0 atau 1, nilai yang ingin diatur untuk b_0

  Lalu, program Anda perlu membaca sebuah baris dengan b_1, status sekarang dari patung yang ada di depan rumah 1; dan keluarkan nilai baru untuk b_1. Hal ini berlanjut untuk setiap dari N rumah. Setelah Anda melewati rumah terakhir (sebagai contoh, Anda membaca dan menulis b_{N-1}) program Anda perlu berhenti.

  Catat bahwa program Anda hanya bisa membaca nilai berikutnya b_{i+1} setelah mengeluarkan nilai b_i$.

- Jika w adalah genap. Detje akan berjalan dari sekolah ke rumahnya, dan dia akan melewati rumah-rumah dengan urutan terbalik $N - 1, N - 2, ...0$.

Proses ini hampir sama dengan ketika w adalah ganjil, kecuali ketika Anda memulai dengan membaca dan menulis b_{N-1}, lalu b_{N-2}, dan seterusnya hingga b_0.

Ketika $w = 1$, masukan $b_0, b_1, ..., b_{N-1}$ berisi status aslinya dari patung-patung burung. Ketika $w > 1$, masukan $b_0, b_1, ..., b_{N-1}$ ke program Anda berdasarkan hasil dari eksekusi terakhir program Anda.

Terakhir, setelah eksekusi terakhir program Anda, nilai dari b_i perlu sama dengan nilai awal b_{a_i} untuk semua i. Jika selain itu, maka Anda akan mendapatkan hasil "Wrong Answer".

Detail

Jika *total* dari semua waktu eksekusi dari $W + 1$ eksekusi terpisah dari program Anda melebihi batas waktu, *submission* Anda akan mendapatkan hasil "Time Limit Exceeded".

Pastikan untuk melakukan *flush* ke *standard output* setelah setiap baris yang program Anda keluarkan, atau program Anda akan mendapatkan hasil "Time Limit Exceeded". Di Python, hal ini akan terjadi secara otomatis selama Anda menggunakan `input()` untuk membaca baris. Di C++, `cout << endl;` melakukan *flush* dengan tambahan sebuah baris baru; gunakan `fflush(stdout)` untuk printf.

Batasan dan Penilaian

- $2 \leq N \leq 500$.
- Anda dapat menggunakan paling banyak $W \leq 60$ ronde.

Solusi Anda akan diuji dengan sekumpulan kasus uji, yang setiapnya bernilai sejumlah poin. Setiap *test group* berisi sekumpulan kasus uji. Untuk mendapatkan poin pada suatu *test group*, Anda perlu menyelesaikan seluruh kasus uji yang ada di *test group* tersebut.

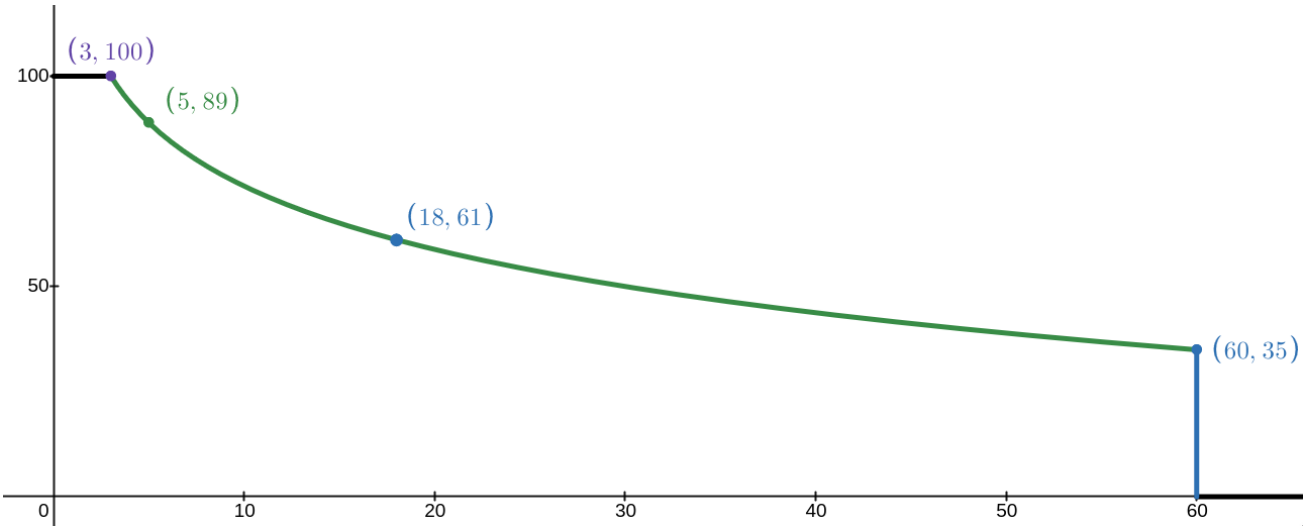
Grup	Nilai Maksimum	Batasan
1	10	$N = 2$
2	24	$N \leq 15$
3	9	$a_i = N - 1 - i$
4	13	$a_i = (i + 1) \bmod N$
5	13	$a_i = (i - 1) \bmod N$
6	31	No additional constraints

Untuk setiap *test group* yang program Anda telah selesaikan dengan benar, Anda akan mendapatkan nilai berdasarkan formula berikut:

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

dengan S_g adalah nilai maksimum untuk *test group* tersebut, dan W_g adalah nilai maksimum dari W yang dipakai untuk kasus uji apa pun pada *test group* tersbeut. Nilai Anda akan dibulatkan ke bilangan bulat terdekat.

Grafik di bawah ini menunjukkan sejumlah poin, sebagai fungsi dari W, yang akan didapat oleh program Anda jika menyelesaikan seluruh *test group* dengan W dengan nilai sama. Secara khusus, untuk mendapatkan nilai 100 di soal ini, Anda perlu menyelesaikan semua kasus uji dengan $W \leq 3$.



Testing Tool

Untuk memfasilitasi pengujian solusi Anda, kami telah menyiapkan sebuah alat sederhana yang Anda bisa unduh. Lihat "attachments" di bagian bawah halaman soal Kattis. Penggunaan alat ini opsional. Catat bahwa *grader* resmi yang terdapat pada Kattis akan berbeda dengan alat pengujian ini.

Untuk menggunakan alat ini, buatlah sebuah *file* Masukan, seperti "sample1.in", yang perlu dimulai dengan bilangan N dan diikuti oleh sebuah baris dengan N bilangan yang menunjukan permutasi, dan baris lainnya dengan N bit (0 or 1) yang menentukan keadaan awal patung burung. Sebagai contoh:

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

Untuk program Python, sebut `solution.py` (biasa dijalankan sebagai `pypy3 solution.py`):

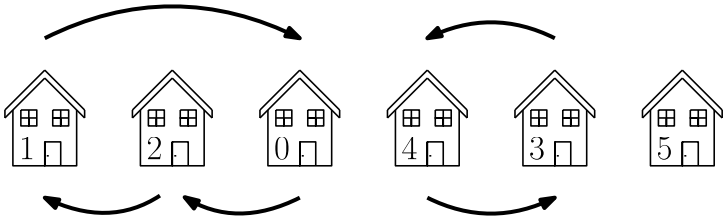
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Untuk program C++, pertama *compile* programmnya (misalnya dengan `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) lalu jalankan:

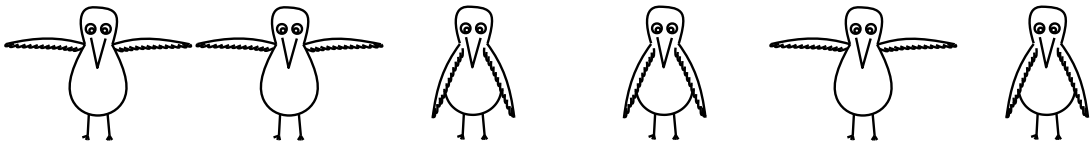
```
python3 testing_tool.py ./solution.out < sample1.in
```

Contoh

Pada contoh berikut, kita diberikan permutasi berikut untuk warga yang ada di rumah:

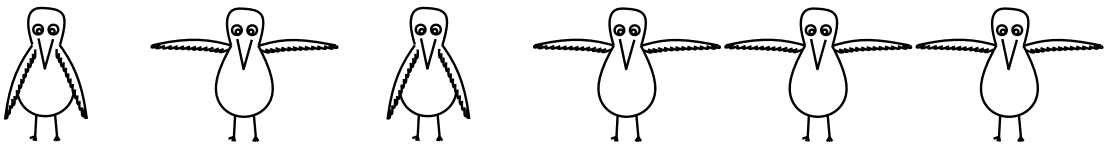


Pertama kalinya contoh program dieksekusi (dengan $w = 0$), programnya akan menjalankan $W = 2$, yang berarti Detje akan berjalan sepanjang jalan sebanyak dua kali (dan program akan dijalankan dua atau lebih kali). Sebelum perjalanan eprtama, patung-patung burung akan seperti berikut:



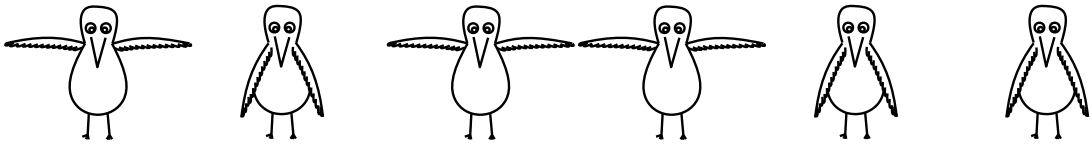
Lalu program akan dieksekusi dengan $w = 1$: mengindikasikan perjalanan pertama Detje. Ia melewati setiap patung burung satu per satu, mulai dari kiri, dan bisa jadi menyesuaikan status beberapa patungnya. Contoh program perlu mengeluarkan status dari patung burung ke-i sebelum kita dapat melihat burung ke $i + 1$.

Setelah Detje sampai di sekolah, status dari patung-patung burung akan menjadi seperti ini:



Pada eksekusi terakhir program (dengan $w = 2$), Detje berjalan balik ke rumah dari sekolahnya. Ingat bahwa di kasus ini, ia akan berjalan melalui patung-patung burung dari kanan ke kiri dan memprosesnya dengan urutan yang terbalik! Hal ini berarti ia perlu menentukan status dari patung burung ke-i sebelum dapat melihat burung ke-$i - 1$.

Setelah ia menyelesaikan perjalanannya, status dari patung-patung burung akan menjadi seperti ini:



Memang, ini adalah konfigurasi yang benar. Sebagai contoh, patung burung 3 (keempat dari kiri) sekarang terbuka (now $b_3 = 1$), yang adalah benar karena warga 4 akan pindah ke sana ($a_3 = 4$) dan ia pada awalnya memiliki patung burung dengan sayap terbuka (nilai awal $b_4 = 1$).

Keluaran *grader*	Keluaran Anda
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