

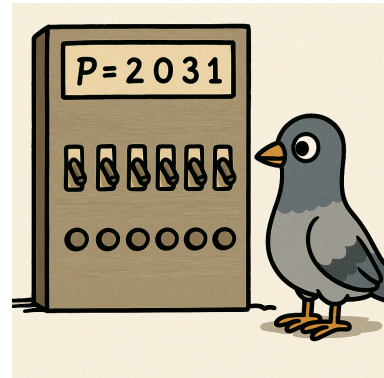


PROBLEMA GUESS THE PERMUTATION

Guguștiucul Gimi a ajuns într-o situație destul de complicată!

El a găsit o mașinărie veche. Înăuntrul acestei mașinării există N întrerupătoare ascunse $s[0], s[1], s[2], \dots, s[N-1]$. Orice întrerupător i poate să fie ori închis (reprezentat de $s[i] = 0$) ori deschis (reprezentat de $s[i] = 1$). Inițial, toate întrerupătoarele sunt închise. Totodată, mașinăria are N butoane numerotate de la 0 la $N-1$, un ecran, și o permutare secretă $P = p[0], p[1], p[2], \dots, p[N-1]$ de numere $0, 1, 2, \dots, N-1$.

Un poster mare de pe mașinărie spune: "Dacă poți ghici permutarea P , vei câștiga un premiu mare!" Acum Gimi vrea premiul, așa că vă cere ajutorul.



Situație complicată

Pentru a determina P , voi puteți apăsa butoanele de pe mașinărie. Când apeși al i -lea buton, se întâmplă următoarele:

1. Fie k numărul de apăsări anterioare ale oricărui buton.
2. Întrerupătorul cu indexul $p[k \% N]$ este schimbat. (formal, $s[p[k \% N]] := 1 - s[p[k \% N]]$);
3. Ecranul afișează starea curentă a întrerupătorului cu indexul i (valoarea lui $s[i]$).

CERINȚĂ Vi se dă numărul întreg N . Trebuie să determinați permutarea ascunsă P , apăsând butoanele mașinăriei de un număr suficient de mic de ori.

Deoarece mașinăria este veche, se va strica dacă apăsați butoanele de prea multe ori, așa că, dacă butoanele au fost apăstate de mai mult de $50 \cdot N$ ori, nu primiți premiul.

Din aceleași motive, doriți să reduceți la minimum numărul total de apăsări ale butoanelor. Pentru restricții mai detaliate, consultați secțiunea Punctaj.

IMPLEMENTARE Fișierul trimis trebuie să conțină `#include "permutation.h"`.

Ar trebui să implementați următoarea funcție:

```
std::vector<int> solve(int N);
```

Această funcție primește N ca parametru și trebuie să returneze un vector q de lungime N astfel încât pentru orice i ($0 \leq i \leq N-1$), este adevărată relația $q[i] = p[i]$. Această funcție este apelată exact o singură dată pentru fiecare test.

În soluția voastră, puteți apela funcția

```
int press_button(int i);
```

Această funcție returnează rezultatul apăsării celui de-al i -lea buton, adică valoarea $s[i]$ după pașii 2 și 3 de mai sus.

RESTRICȚII

- ◆ $N \leq 10^4$
- ◆ P este o permutare a numerelor $0, 1, 2, \dots, N-1$.



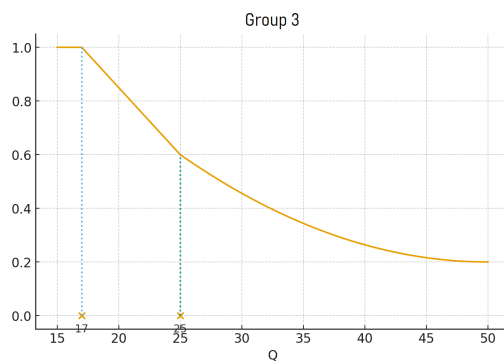
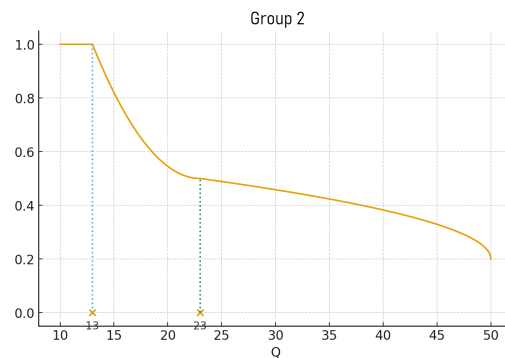
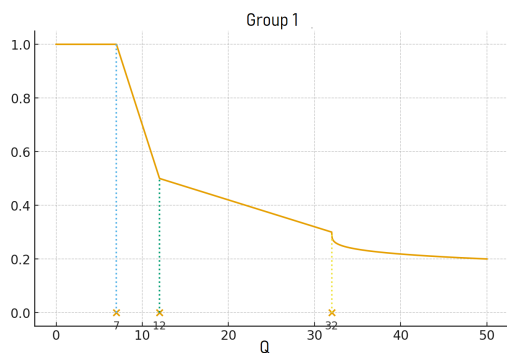
#	Puncte	Restricții
1	20	$N = 32$
2	40	$N = 1\,000$
3	40	$N = 10\,000$

PUNCTAJ Fiecare test este evaluat după cum urmează: Fie K numărul de apelări ale funcției `press_button()` făcute de soluția ta. Fie $Q = \frac{K}{N}$. Dacă permutarea este ghicită greșit sau dacă $Q > 50$, atunci scorul este 0. Altfel, scorul este dat de S înmulțit cu scorul maxim al grupului, unde S este definit după cum urmează:

$$\text{Group 1. } S = \begin{cases} 1, & Q \leq 7 \\ 1 - \frac{1}{10}(Q - 7), & 7 < Q \leq 12 \\ \frac{1}{2} - \frac{1}{100}(Q - 12), & 12 < Q \leq 32 \\ \frac{3}{10} - \frac{1}{10}\sqrt[4]{\frac{Q-32}{18}}, & 32 < Q \leq 50 \end{cases}$$

$$\text{Group 2. } S = \begin{cases} 1, & Q \leq 13 \\ \frac{1}{2} + \frac{1}{2} \left(\frac{23-Q}{10} \right)^2, & 13 < Q \leq 23 \\ \frac{1}{5} + \frac{3}{10} \sqrt{\frac{50-Q}{27}}, & 23 < Q \leq 50 \end{cases}$$

$$\text{Group 3. } S = \begin{cases} 1, & Q \leq 17 \\ \frac{3}{5} + \frac{25-Q}{20}, & 17 < Q \leq 25 \\ \frac{1}{5} + \frac{2}{5} \left(\frac{50-Q}{25} \right)^2, & 25 < Q \leq 50 \end{cases}$$





■ **EXEMPLU DE INTERACȚIUNE** Se consideră un test cu $N = 4$ și $P = [1, 3, 0, 2]$ și următoarele apeluri ale `press_button()`.

Action	S	Return Value
<code>press_button(0)</code>	0100	
		0
<code>press_button(1)</code>	0101	
		1
<code>press_button(2)</code>	1101	
		0
<code>press_button(3)</code>	1111	
		1
<code>press_button(1)</code>	1011	
		0
<code>press_button(3)</code>	1010	
		0
<code>press_button(2)</code>	0010	
		1
<code>return {1,3,0,2}</code>		

■ **GRADER LOCAL** Vi se oferă un exemplu de evaluator (`sample-grader.cpp`).
Evaluatorul citește datele de intrare în următorul format:

- linia 1: un număr întreg N
- linia 2: N numere întregi $p[0], p[1], p[2], \dots, p[N - 1]$.

Apoi apelează funcția `solve(N)` și afișează verdictul soluției tale împreună cu numărul de interogări utilizate sau explicația erorii.