

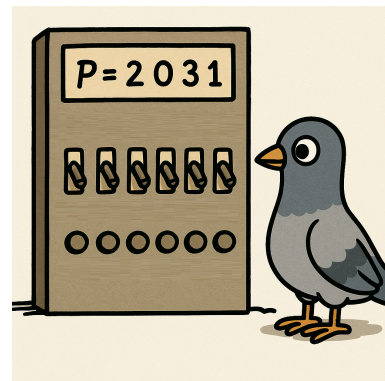


PROBLEM GUESS THE PERMUTATION

Gimi the Gugustiuc has ended up in a pretty complicated situation!

He had found an old machine. Inside the machine there are N hidden switches $s[0], s[1], s[2], \dots, s[N - 1]$. Any switch i can be either turned off (represented by $s[i] = 0$) or turned on (represented by $s[i] = 1$). Initially, all switches are turned off. The machine also has N buttons numbered from 0 to $N - 1$, a screen, and a secret permutation $P = p[0], p[1], p[2], \dots, p[N - 1]$ of numbers $0, 1, 2, \dots, N - 1$.

A big poster on the machine says: “if you can guess the permutation P , you win a big prize!” Now Gimi wants the prize, so he asks you for help.



Complicated situation

To determine P , you can press the buttons on the machine. When you press the i -th button, the following happens:

1. Let k be the number of times you have pressed any button before this.
2. The switch with index $p[k \% N]$ is flipped (formally, $s[p[k \% N]] := 1 - s[p[k \% N]]$);
3. The screen displays the current state of the switch with the index i (the value of $s[i]$).

- **TASK** You are given the integer N . You must determine the hidden permutation P , by pushing the buttons of the machine a small enough number of times.

Since the machine is old, it will break if you press the buttons too many times, so if the buttons were pressed more than $50 \cdot N$ times, you do not get the prize.

For the same reasons, you want to minimize the total number of times you press the buttons. For more detailed constraints, please refer to the Scoring section.

- **IMPLEMENTATION** The submitted file must contain `#include "permutation.h"`.

You should implement the following function:

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

This function receives N as a parameter and must return a vector q of size N such that for every $0 \leq i \leq N - 1$, it holds that $q[i] = p[i]$. This function is called exactly once for each test case.

In your solution, you can call the function

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

This function returns the result of pushing the i -th button, i.e. the value of $s[i]$ after steps 2 and 3 above.

- **RESTRICTIONS**
 - ◆ $N \leq 10^4$
 - ◆ P is a permutation of numbers $0, 1, 2, \dots, N - 1$.



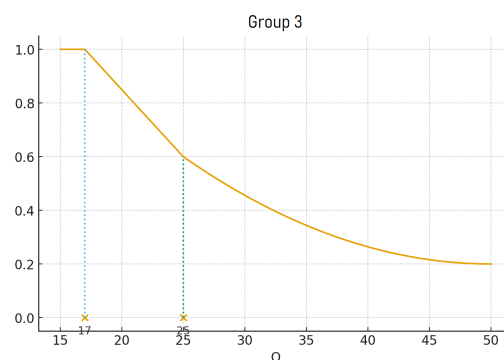
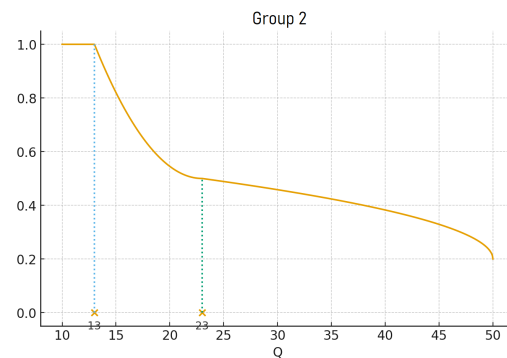
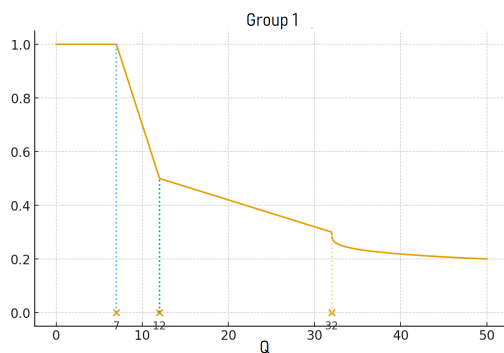
#	Points	Constraints
1	20	$N = 32$
2	40	$N = 1\ 000$
3	40	$N = 10\ 000$

■ **SCORING** Each test case is scored as follows: Let K be the number of times function `press_button()` was called by your solution. Let $Q = \frac{K}{N}$. If the permutation is guessed incorrectly, or if $Q > 50$, then the score is 0. Otherwise, the score is given by S times the maximum group score, where S is defined as follows:

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

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

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





- **INTERACTION EXAMPLE** Consider a test case with $N = 4$ and $P = [1, 3, 0, 2]$, and the following calls to `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>		

- **SAMPLE GRADER** You are provided with a sample grader (`sample-grader.cpp`). The grader reads input in the following format:

- line 1: integer N
- line 2: N integers $p[0], p[1], p[2], \dots, p[N - 1]$.

It then calls the function `solve(N)` and outputs the verdict of your solution together with the number of queries used or explanation of the failure.