

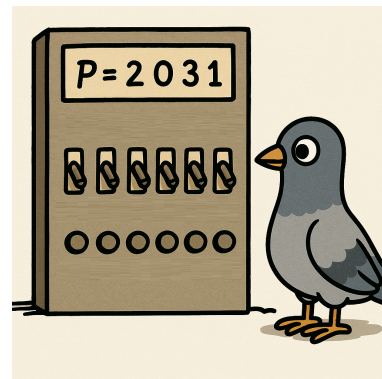


Задатак Погоди пермутацију

Гими Гугустиук се нашао у прилично компликованој ситуацији!

Пронашао је стару машину. Унутар машине налази се N скривених прекидача $s[0], s[1], s[2], \dots, s[N-1]$. Сваки прекидач i може бити или искључен (означено са $s[i] = 0$) или укључен (означено са $s[i] = 1$). У почетку су сви прекидачи искључени. Машина такође има N дугмади нумерисаних бројевима између 0 и $N-1$, екран и тајну пермутацију $P = p[0], p[1], p[2], \dots, p[N-1]$ бројева $0, 1, 2, \dots, N-1$.

На великом постеру на машини пише: „ако успете да погодите пермутацију P , освајате велику награду!“ Сада Гими жели награду, па тражи вашу помоћ.



Complicated situation

Да бисте одредили P , можете притискати дугмад на машини. Када притиснете i -то дугме, дешава се следеће:

- Нека је k број пута када сте до тада притиснули било које дугме.
- Прекидач са индексом $p[k \% N]$ се обрће (формално, $s[p[k \% N]] := 1 - s[p[k \% N]]$);
- На екрану се приказује тренутно стање прекидача са индексом i (вредност $s[i]$).

Задатак Задато вам је целобројно N . Морате да одредите скривену пермутацију P , притискајући дугмад на машини довољно мали број пута.

Пошто је машина стара, поквариће се ако предуго притискате дугмад, па ако су дугмад притиснута више од $50 \cdot N$ пута, не добијате награду.

Из истих разлога, желите да минимизирате укупан број притисака дугмади. За детаљнија ограничења погледајте одељак о оцењивању.

Имплементација Послати фајл мора да садржи `#include "permutation.h"`.

Потребно је да имплементирате следећу функцију:

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

Ова функција добија N као параметар и мора да врати вектор q дужине N такав да за свако $0 \leq i \leq N-1$ важи $q[i] = p[i]$. Ова функција се позива тачно једном за сваки тест пример.

У вашем решењу можете да позивате функцију

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

Ова функција враћа резултат притиска i -тог дугмета, тј. вредност $s[i]$ након корака 2 и 3 описаних изнад.



Ограничења

- ◆ $N \leq 10^4$
- ◆ P је пермутација бројева од 0 до $N - 1$

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

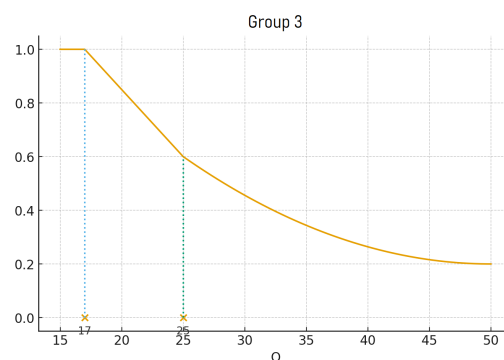
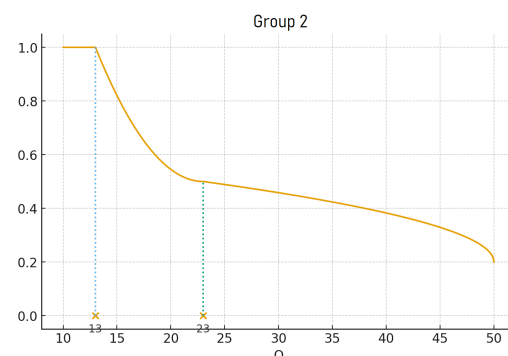
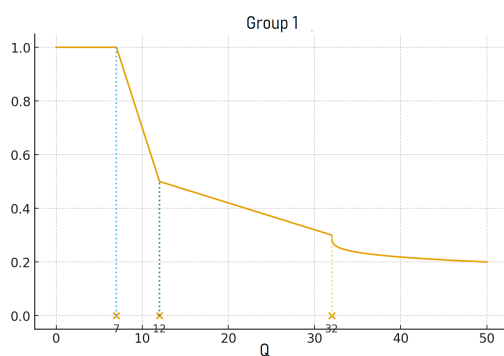
Оцењивање

Сваки тест пример се оцењује на следећи начин: Нека је K број пута када је функција `press_button()` позвана у вашем решењу. Нека је $Q = \frac{K}{N}$. Ако је пермутација погрешно погођена или ако је $Q > 50$, добијате 0 поена. У супротном, резултат се добија као S помножен са максималним бројем поена за групу, где је S дефинисан на следећи начин:

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

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

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





- **Пример интеракције** Размотримо тест пример са $N = 4$ и $P = [1, 3, 0, 2]$, и следеће позиве функције `press_button()`.

Акција	S	Повратна вредност
<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>		

- **Пример грејдера** Добијате пример грејдера (`sample-grader.cpp`).
Грејдер чита улаз у следећем формату:

- ред 1: цео број N
- ред 2: N целих бројева $p[0], p[1], p[2], \dots, p[N - 1]$.

Затим позива функцију `solve(N)` и исписује пресуду за ваше решење заједно са бројем коришћених упита или објашњењем грешке.