

# Zgadywanie permutacji (c)

Limit pamięci: 128 MB

Limit czasu: 5.00 s

To zadanie jest interaktywne. Po wypisaniu każdego wiersza należy opróżnić bufor. W C++ możesz użyć `cout << flush`, w Java – `System.out.flush()`, w Python – `sys.stdout.flush()`. Należy ściśle trzymać się protokołu opisanego w rozdziale *Protokół interakcji* – niewczytanie wiadomości wysłanej przez interaktor skutkuje werdyktem *wrong answer*.

Ukryta jest pewna permutacja liczb od 1 do  $n$ . Można, dla dowolnych  $1 \leq l \leq r \leq n$  oraz  $1 \leq x \leq n$  zadawać pytania postaci: czy pomiędzy  $l$ -tym a  $r$ -tym elementem (włącznie) tej permutacji jest pewien element podzielny przez  $x$ ? Można zadać co najwyżej  $Q$  zapytań. W odpowiedzi należy wypisać tę ukrytą permutację.

## Interakcja

Na początku należy wczytać 1 liczbę całkowitą  $n$ , oznaczającą długość permutacji. Po zadaniu zapytania postaci `? l r x`, należy wczytać liczbę całkowitą, 0, jeżeli odpowiedź na to zapytanie jest negatywna oraz 1 w przeciwnym wypadku.

Na końcu należy wypisać znalezioną permutację  $a$  w formacie: `! a_1 a_2 ... a_n`.

## Podzadania

We wszystkich podzadaniach zachodzi:  $1 \leq n \leq 10\,000$ ,  $Q = 500\,000$ .

| T | Warunki                                  | Punkty |
|---|------------------------------------------|--------|
| 1 | $n = 10$                                 | 5      |
| 2 | $n = 1000$                               | 15     |
| 3 | Gradacja względem liczby zadanych pytań. | 80     |

Dla podzadania 3 gradacja wygląda w następujący sposób: niech  $z$  oznacza liczbę zadanych przez program pytań, wtedy procent punktów przyznanych za dany test wynosi:

- 100, jeżeli  $z \leq 135\,000$ ,
- $40 + 60 \cdot \frac{250\,000 - z}{250\,000 - 135\,000}$ , gdy  $135\,000 < z \leq 250\,000$
- $40 \cdot \frac{500\,000 - z}{500\,000 - 250\,000}$ , gdy  $250\,000 < z \leq 500\,000$
- 0 gdy  $z > 500\,000$ .

## Przykładowa interakcja

Wejście Wyjście

```
5
? 5 5 5
1
? 2 3 4
1
! 3 1 4 2 5
```

## Narzędzia do testowania lokalnego

Zostaną udostępnione testy przykładowe i programy `zgasoc.cpp`, `run.py` oraz przykładowe błędne rozwiązanie o poprawnym schemacie komunikacji `zga.cpp`. Żeby urochomić rozwiązanie na teście należy użyć komendy:

```
python3 run.py 1 [plik wykonywalny z zgasoc.cpp] [plik wykonywalny z rozwiązaniem]
```

[test]

Na przykład: `python3 run.py 1 ./kulsoc ./kul 0a`