

# Turniej lockouta

smol PREOI 2024  
Dzień 3

Kod zadania: **loc**  
Limit pamięci: **128 MiB**



Właśnie zakończył się największy, najświetniejszy turniej Lockout'a w historii informatyki. Na nieszczęście, pewien człowiek potknął się o pewien kabel, co wyłączyło pewien serwer, co tak się składa, wyczyściło tablicę wyników. Przybyłeś na miejsce po fakcie i aby wręczyć nagrody, musisz teraz ustalić zwycięzcę. W turnieju brało udział  $2^n$  uczestników i przebiegł on następująco: w 1. turze uczestnik 1. grał z 2., 3. grał z 4., 5. z 6. itd., tak że z każdej rozgrywki wyszedł dokładnie jeden zwycięzca. Tych  $2^{n-1}$  zwycięzców walczyło w 2. turze w sposób analogiczny (zwycięzca meczu 1. vs 2. ze zwycięzcą 3. vs 4., zwycięzca 5. vs 6. ze zwycięzcą 7. vs 8. itd.), po czym zwycięzcy przeszli do 3. tury. Po  $n$  turach pozostał dokładnie jeden zawodnik, którego to właśnie szukasz.

Zebrałeś wszystkich  $2^n$  uczestników do jednego pokoju, zamknąłeś drzwi na klucz i nie wypuścisz ich dopóki nie poznasz zwycięzcy. Niestety informatycy nie współpracują, więc niczego nie dowiesz się od nich wprost. Trzeba będzie zastosować techniki psychologiczne, oraz trochę dedukcji. Twoja strategia wygląda następująco: wypowiadasz na głos dwa nazwiska, obserwujesz twarze ich posiadaczy, po czym zawsze dasz radę ustalić, który z nich wygrał więcej meczów, lub też że wygrali dokładnie tyle samo, jeśli tak jest. Chciałbyś ustalić zwycięzcę za pomocą jak najmniejszej liczby pytań, przy czym liczba  $\frac{1}{3}2^{n+1}$  cię w zupełności satysfakcjonuje.

## Komunikacja

Twój program będzie korzystał z dostarczonej biblioteki odpowiadającej na podane pytania. Aby użyć biblioteki, należy wpisać w swoim programie:

```
#include "loc_lib.h"
```

Biblioteka udostępnia następujące funkcje:

- `int` `daj_n()` – Funkcja daje w wyniku jedną liczbę całkowitą  $n$  ( $0 \leq n \leq 20$ ). Jest  $2^n$  uczestników turnieju.
- `char` `pytanie(int a, int b)` – Funkcja zwraca '`<`' jeśli  $b$  wygrał więcej meczów, '`>`', jeśli  $a$  wygrał więcej i '`=`' jeśli wygrali tyle samo.
- `void` `odpowiedz(int r)` – Funkcja kończy wykonanie programu, a jeśli  $r$  nie jest numerem zwycięzcy, program otrzyma werdykt *Błędna odpowiedź*.

Program nie może otwierać żadnych plików, ani używać standardowego wejścia i wyjścia. Może używać wyjścia diagnostycznego (`stderr`), ale tak jakby po co.

## Eksperymenty

W zakładce Pliki znajduje się folder `dłazaw`, a w nim przykładowe pliki `loc_lib.h` i `loc_lib.cc`. Nie są to biblioteki, które będą użyte do oceny twojego zgłoszenia, ale możesz ich użyć do testowania swojego rozwiązania, jeśli umieścisz je w tym samym folderze i do komendy kompilacji dodasz na końcu `loc_lib.cc`. Rozwiązanie skompilowane z tymi bibliotekami przyjmuje jako standardowe wejście  $n$  linijek, pierwsza definiująca  $n$ , a każda  $i$ -ta kolejna wymieniająca w kolejności rosnącej zwycięzców  $i$ -tej rundy.

## Przykład

Wejście dla testu `loc0a`:

```
4
1 4 5 7 10 12 14 15
1 5 12 15
5 12
12
```

Możliwy przebieg programu:

Wywołanie	Wynik
<code>daj_n()</code>	4
<code>pytanie(1, 2)</code>	<code>&gt;</code>
<code>pytanie(1, 5)</code>	<code>&lt;</code>
<code>pytanie(1, 15)</code>	<code>=</code>
<code>pytanie(12, 5)</code>	<code>&gt;</code>
<code>pytanie(15, 12)</code>	<code>&lt;</code>
<code>odpowiedz(12)</code>	OK, 5 zapytań.



# Ocenianie

Niech  $p$  będzie liczbą zapytań, jakie wykonał twój program. Wtedy liczbę punktów jakie on otrzyma (chyba że ze wzorku wynika że będzie to liczba ujemna) opisuje tabelka:

Podzadanie	Ograniczenia	Limit czasu	Liczba punktów
1	$0 \leq n \leq 1$	1 s	5
2	$n = 2$	1 s	25 - 5p
3	$3 \leq n \leq 4$	10 s	$20 \cdot f(p)$
4	$5 \leq n \leq 10$	5 s	$20 \cdot f(p)$
5	$11 \leq n \leq 20$	5 s	$40 \cdot f(p)$

Gdzie

$$f(x) = \min \left( 100, \left\lfloor \frac{100 \cdot 2^n - 20 \lceil \frac{1}{3} 2^{n+1} \rceil - 80x}{2^n - \lceil \frac{1}{3} 2^{n+1} \rceil} \right\rfloor \right) \%$$

(Innymi słowy jest to funkcja liniowa przyjmująca  $f(2^n) = 20\%$ ,  $f(\lceil \frac{1}{3} 2^{n+1} \rceil) = 100\%$ .)