

# Zadanie: CHO

## Chochlik



POTYCZKI ALGORYTMICZNE

Potyczki Algorytmiczne 2018, wielki finał. Limity: 256 MB, 1 s.

13.01.2019

Czarnoksiężnik Bajtazar odnalazł wreszcie, po wielu latach, tekst rytuału, za pomocą którego może przywołać Wielkiego Chochlika Binarnego, aby ten odpowiedział mu na odwieczne pytanie ludzkości: *Czy  $P = NP$ ?* Aby jednak dopełnić rytuał, konieczne jest wyrecytowanie pełnego imienia Chochlika, którego Bajtazar nie zna. Wiadomo tylko, że imię jest słowem składającym się z dokładnie 100 znaków. Jak na Binarnego Chochlika przystało, są to wyłącznie znaki 0 i 1.

Bajtazar może próbować rytuału wielokrotnie (nie więcej jednak niż 222 razy, po których jego stara alchemiczna aparatura ostatecznie odmówi posłuszeństwa). Jeśli choć raz wypowie prawidłowe imię Chochlika, rytuał powiedzie się, a światowa informatyka zmieni się na zawsze. Dzięki swojej magicznej intuicji Bajtazar będzie po każdej próbie dokładnie wiedział, ile znaków z imienia Chochlika tym razem wypowiedział poprawnie. Jest jednak dodatkowa komplikacja: Chochlik jest niezwykle kapryśnym stworzeniem i czasem zmienia zdanie odnośnie tego, jak sam ma na imię. Dokładniej mówiąc, po każdej **nieudanej** próbie Bajtazara Chochlik losuje liczbę  $k$  ze zbioru  $\{0, 1, \dots, 99\}$  z rozkładem jednostajnym. Jeśli okaże się, że  $k$ -ty znak został przez Bajtazara w tej próbie wypowiedziany poprawnie, Chochlik natychmiast zmieni go w swoim imieniu na przeciwny (0 na 1, a 1 na 0).

Jako pechowy uczeń czarnoksiężnika musisz pomóc mu odgadnąć imię Chochlika – zrób to szybko, inaczej wściekły Bajtazar może zamienić Cię w ...no, w coś paskudnego. Na przykład w hydrant. I ciesz się, że akurat nie próbujecie przywołać Bardzo Wielkiego Chochlika Szesnastkowego.

## Komunikacja z biblioteką

Aby móc rozwiązywać zadanie, należy na początku programu dopisać dyrektywę

```
#include "cho.h"
```

Biblioteka dostarcza tylko jedną funkcję:

```
int recytuj(const std::string& proba);
```

która działa w sposób opisany powyżej – jeśli ciąg `proba` jest identyczny z prawdziwym imieniem Chochlika, Twój program natychmiast kończy działanie i zostaje na tym teście zaakceptowany. Jeśli nie – funkcja zwraca liczbę (między 0 a 99) równą liczbie znaków zgodnych z prawdziwym imieniem, a dodatkowo wykonywany jest fragment kodu podobny do następującego:

```
int k = rand() % 100; // Prawdziwa implementacja biblioteki nie używa funkcji rand().
if(proba[k] == prawdziwe_imie[k])
    flip(prawdziwe_imie[k]);
```

gdzie `prawdziwe_imie` oznacza (tajne) imię chochlika, a `flip()` zmienia odpowiedni znak z 0 na 1 i odwrotnie.

Przy każdym zapytaniu, `proba` musi być ciągiem składającym się z dokładnie 100 znaków, każdy 0 lub 1. Nie możesz też wywołać funkcji `recytuj()` więcej niż 222 razy – jeśli złamiesz którąś z tych reguł, dostaniesz komunikat *Błędna Odpowiedź*.

Początkowe imię chochlika jest ustalone na początku każdego testu.

## Przykładowa biblioteka

Do Twojej dyspozycji w dziale Pliki w SIO jest dostępna paczka z przydatnymi plikami, zawierająca przykładową bibliotekę `chozaw.cpp`, która implementuje podaną wyżej funkcję.

Aby skompilować Twój program (powiedzmy, `cho.cpp`) z tą biblioteką, wpisz następującą komendę:

```
g++ cho.cpp chozaw.cpp -o cho -std=c++11 -O2
```

## Wejście dla przykładowej biblioteki

Biblioteka przyjmuje wejście w następującej formie:

W pierwszym i jedynym wierszu powinno się znaleźć imię chochlika, składające się z 100 znaków 0 lub 1.

## Przykład

Dla uproszczenia, załóżmy że imię chochlika ma długość 6. To założenie dotyczy tylko tego przykładu i taki test nie pojawi się w zestawie testowym (wysyłane rozwiązania nie będą na nim uruchamiane). Niech imię chochlika będzie 42 (tj. 101010). Powiedzmy, że wylosowana pozycja  $k$  to 3 w pierwszym wywołaniu i 0 w drugim.

| imię   | wywołanie         | wynik | k | komentarz   |
|--------|-------------------|-------|---|---|
| 101010 | recytuj("010010") | 3     | 3 | Słowa 101010 i 010010 zgadzają się na trzech ostatnich pozycjach, więc zwrócony wynik to 3. Wylosowane $k$ to też 3. Jako że trzeci (licząc od zera) bit zgadzał się w naszym zapytaniu, zostaje on zmieniony w imieniu chochlika, które od teraz brzmi 101110. |
| 101110 | recytuj("001111") | 4     | 0 | Tym razem okazuje się, że pomyliliśmy się na $k$ -tej pozycji ( $k = 0$ ), więc imię pozostaje bez zmian.   |
| 101110 | recytuj("101110") | 6     | - | Zgadliśmy imię chochlika. Twój program będzie automatycznie zakończony. Nie jest losowane nowe $k$ .  |