

Zadanie: KAB

Kable



Trening przed BOI, dzień pierwszy. Plik źródłowy kab.* Dostępna pamięć: 128 MB. 15.07.2020

Każdy wie, jak wygląda typowa serwerownia. Mrugające lampki, walające się wszędzie pudełka po pizzy, niezidentyfikowane kable, o które zawsze ktoś się potknie. . . Niestety, kierownictwo zarządziło uporządkowanie tego bałaganu. W trakcie przeglądania stanu inwentarza trafień na tajemniczą skrzynkę, z której wystaje n końcówek należących do $n/2$ kabli (dziwnym trafem każdy kabel ma dokładnie dwie końcówki). Niestety, nie wiesz, które spośród końcówek należą do tego samego kabla.

Na szczęście, do pomocy masz n ochotników – krewnych, znajomych i wyznawców. Każdy z nich ma przypisaną jedną z wystających końcówek. Na dany znak niektórzy z nich łapią rękami swoje końcówki, zaś pozostali do swoich podłączają prąd (o stosunkowo niskim napięciu). Jeśli któryś z kabli zostanie z jednej strony podłączony do prądu, a z drugiej złapany, osoba na tym drugim końcu odczuje impuls elektryczny (nie przeszkadzający jednak w dalszym poświęcaniu się dla nauki).

Wydając odpowiednie komendy i obserwując działanie prądu, ustal przynależność końcówek kabli.

Działanie programu

Twój program nie powinien czytać żadnych danych ani wypisywać żadnych danych. Zamiast tego zostanie skompilowany z odpowiednią biblioteką oceniającą. Aby użyć biblioteki, należy wpisać na początku programu:

```
#include "ckablib.h"
```

Biblioteka udostępnia następujące funkcje:

- `int inicjuj()` – zwraca liczbę parzystą $n \leq 1\,000\,000$ – liczbę końcówek kabli. Funkcja ta powinna zostać użyta dokładnie raz, na samym początku działania programu.
- `void podlacz(bool [] napięcie, bool [] kopnieci)` – symuluje jedno podłączenie kabli. Zmienna `napięcie` powinna być tablicą, w której elementy o indeksach od 1 (!) do N odpowiadają czynnościom wykonanym z kolejnymi końcówkami. Wartość `napięcie[i]` powinna być równa `true`, jeśli i -ta końcówka jest podłączana do prądu, `false`, jeśli chwytana ręką. Tablicę `kopnieci[]` wypełnia biblioteka – `kopnieci[i]` otrzymuje wartość `true` wtedy i tylko wtedy, gdy i -ta osoba została porażona prądem.
- `void odpowiedz (int [] koncowki)` – kończy program, udzielając odpowiedzi. Wartości `koncowki[i]` dla $i = 1, 2, \dots, n$ muszą być permutacją zbioru $\{1, 2, \dots, n\}$. Wartość `koncowki[i]` powinna być równa j , jeśli Twoim zdaniem końcówki i oraz j należą do tego samego kabla.

Jakakolwiek komunikacja powinna odbywać się tylko za pośrednictwem wyżej podanych funkcji.

Rozwiązanie będzie kompilowane wraz z biblioteką przy użyciu polecenia:

```
g++ -O2 -static ckablib.c kab.cpp -lm
```

Ocenianie

Twój program będzie oceniany na pewnej liczbie testów. Otrzymana liczba punktów będzie zależała od liczby zapytań (wywołań funkcji `podlacz`). Otrzymasz pełną liczbę punktów za dany test, jeśli zapytań będzie nie więcej niż **30**. W przeciwnym wypadku liczba punktów za ten test będzie dana wzorem:

$$L = 100\% \cdot \frac{1}{\log_{10} N/3}.$$

W szczególności za 300 zapytań otrzymasz połowę punktów, za 3000 jedną trzecią punktów itp. Pamiętaj, że Twój program musi zmieścić się w limicie czasowym.

Przykładowa biblioteka

Do dyspozycji masz przykładową bibliotekę składającą się z plików `ckablib.h` i `ckablib.cpp`. Po skompilowaniu swojego programu wraz z biblioteką podanym wyżej poleceniem otrzymasz wykonywalny program, który:

- Wczytuje dane ze standardowego wejścia: w pierwszym wierszu liczbę końcówek N , w drugim przyporządkowania końcówek: i -ta liczba w tym wierszu to numer końcówki połączonej z i -tą końcówką.
- Rozgrywa komunikację Twojego programu z biblioteką.
- Wypisuje na standardowe wyjście informację, czy Twój program udzielił dobrej odpowiedzi.