



Dostępna pamięć: 256MB

## Osady rzymskie

*Jest rok 50. przed naszą erą. Cała Galia została podbita przez Rzymian...  
Cała? Nie! Jedna, jedyna osada, zamieszkała przez nieugiętych Galów,  
wciąż stawia opór najeźdźcom i uprzykrza życie legionom rzymskim  
stacjonującym w obozach Rabarbarum, Akwarium, Relanium i Delirium...*

Prowadzący inwazję na Galię Rzymianie zaczęli zakładać osady. Ich inwazja rozprzestrzeniała się coraz szybciej. Rosła również liczba stacjonujących tam żołnierzy. Pewnego dnia okazało się, że w osadzie brakuje miejsca na więcej mieszkańców. Ponieważ inwazja miała dopiero nabierać na sile, postanowili założyć nową osadę.

W niej również wkrótce zaczęło brakować miejsca, i w następnej, a także w następnych, a żołnierzy dalej przybywało...

Ponieważ Rzymianie są bardzo oszczędni, budują oni tylko tyle dróg, ile jest im absolutnie niezbędne. Oznacza to, że przy zakładaniu nowej osady jest ona łączona drogą z którąś z istniejących osad, po czym nie są budowane żadne nowe drogi.

Generałowie rzymscy chcą wiedzieć, ile zajmie przejście oddziału między dwoma osadami. Czy potrafisz odpowiedzieć na ich pytania?

### Komunikacja

Napisz bibliotekę komunikującą się z programem oceniającym. Powinna ona zawierać przynajmniej następujące trzy funkcje, wywoływane przez program oceniający:

- `void init(string nazwa)` – ta funkcja zostanie wywołana dokładnie raz, na początku sprawdzania. Możesz ją wykorzystać do inicjalizacji swoich struktur danych. Lub do czegokolwiek innego. Jako parametr funkcja przyjmuje nazwę pierwszej osady.
- `void nowa_osada(string nowa, string stara, int dlugosc)` – wywołanie tej funkcji oznacza, że mieszkańców było na tyle dużo, że założyli nową osadę o nazwie `nowa` i połączyli ją z osadą o nazwie `stara` drogą o długości `dlugosc`. Długości dróg nie przekraczają 1000. Możesz założyć, że osada `stara` została już kiedyś założona (za pomocą wywołania funkcji), a `nowa` jeszcze nie.
- `int odleglosc(string pierwsza, string druga)` – zapytanie generała. Jako odpowiedź należy podać długość najkrótszej ścieżki między osadami `pierwsza` i `druga`.

Twoja biblioteka **nie może** czytać żadnych danych (ani ze standardowego wejścia, ani z plików). **Nie może** również nic wypisywać do plików ani na standardowe wyjście. Może pisać na standardowe wyjście diagnostyczne (`stderr`) - pamiętaj jednak, że zużywa to cenny czas.

Twoja biblioteka **nie może** zawierać funkcji `main`, może natomiast zawierać cokolwiek innego. W szczególności **musi** zawierać podane powyżej funkcje.

### Ograniczenia

Nazwy osad składają się z maksymalnie 10 wielkich liter alfabetu angielskiego. Żadna funkcja nie zostanie wywołana więcej niż 100 000 razy.

## Kompilacja

Twój program zostanie skompilowany za pomocą polecenia:

```
g++ -O2 -static -lm osa.cpp osamain.cc -o osa
```

## Przykład

Wywołanie	Wynik	Komentarz
<code>init("RABARBARUM")</code>	–	Pierwszą osadą w historii jest Rabarbarum.
<code>nowa_osada("AKWARIUM", "RABARBARUM", 3)</code>	–	Do Rabarbarum dołącza Akwarium drogą o długości 3.
<code>odleglosc("AKWARIUM", "RABARBARUM")</code>	3	Wyszło 3, jak nie liczyć.
<code>nowa_osada("RELANIUM", "RABARBARUM", 2)</code>	–	Do Rabarbarum dołącza Relanium, do którego podłączają się
<code>nowa_osada("GALIUM", "RELANIUM", 1)</code>	–	Galium i Relanium
<code>nowa_osada("DELIRIUM", "RELANIUM", 4)</code>	–	
<code>odleglosc("AKWARIUM", "DELIRIUM")</code>	9	Akwarium–Rabarbarum + Rabarbarum–Relanium + Relanium–Delirium
<code>odleglosc("AKWARIUM", "AKWARIUM")</code>	0	...

## Ocenianie

Podzadanie	Ograniczenia	Punkty
1	Każda funkcja będzie wywołana maksymalnie 1000 razy	26
2	Funkcja <code>odleglosc</code> będzie wywołana co najwyżej 100 razy	13
3	Funkcja <code>odleglosc</code> będzie wywołana po ostatnim wywołaniu funkcji <code>nowa_osada</code>	28
4	Brak dodatkowych ograniczeń	33

## Testowanie

W zakładce "pliki" znajduje się przykładowy program oceniający oraz przykładowe rozwiązanie. Rozwiązanie jest błędne, aczkolwiek poprawne technicznie.