# Zadanie: POR
# Portale

**ONTAK 2021, dzień szósty. Dostępna pamięć: 512 MB. Limit czasu: 10 s.** *04.07.2021*

Bajtocja jest największym mocarstwem międzyplanetarnym nowej ery. Jej terytorium rozciąga się na $N$ planet, a strefa wpływów ma zasięgi daleko wykraczające poza Drogę Mleczną. Ale zarządzanie tak rozległym mocarstwem nie jest ani proste, ani tanie, o czym wie Bajtazar – obecny imperator Bajtocji.

Aby ułatwić sobie sprawowanie władzy Bajtazar postanowił wykorzystać najnowszą bajtocką technologię *portali czasoprzestrzennych*. Każdy portal łączy dwie planety i pozwala na szybkie, tanie i obustronne przemieszczanie się między nimi – choć same portale tanie nie są. Istotnie, budżet Bajtocji pozwala na wybudowanie maksymalnie $N - 1$ takich portali, i oczywiście Bajtazar pragnie, aby po ich zbudowaniu między każdymi dwiema planetami istniało (niekoniecznie bezpośrednie) połączenie. Jednak Bajtazar nie będzie zadowolony byle jaką siecią połączeń – żąda, aby po rozmieszczeniu wspomnianych $N - 1$ par portali **ścieżka między dowolnymi dwiema planetami miała długość nie przekraczającą $D$ skoków przez portale**.

To wszystkie wymogi Bajtazara, ale dopiero początek trudności. Otóż portale czasoprzestrzenne mogą zostać wybudowane między wybraną parą planet jedynie w czasie tzw. *załamania czasoprzestrzennego*. Załamanie czasoprzestrzenne to chwilowa wyrwa w prawach fizyki, która zawsze prowadzi między dwoma różnymi masywnymi obiektami (na przykład planetami). Portal między dwiema planetami można zbudować tylko utrwalając właśnie istniejące załamanie czasoprzestrzenne.

Bajtoccy naukowcy przewidują, że w niedalekiej przyszłości nastąpi $M$ załamań czasoprzestrzennych, każde między pewną parą planet – wtedy właśnie powstanie sieć żądana przez imperatora Bajtazara. Nikt jednak nie wie, które planety dokładnie zostaną tymczasowo połączone przez załamania czasoprzestrzenne – wiadomo tylko, że **każde nowe załamanie ma jednakową szansę pojawienia się między każdą z $\binom{n}{2}$ par różnych planet**. Załamania trwają bardzo krótko – zanim pojawi się następne, poprzednie znika, o ile nie zostanie utrwalone przez budowę portalu. Jednak istniejące w przeszłości załamanie nie wpływa na przyszłe prawdopodobieństwa, i zawsze ma szansę pojawienia się ponownie.

Aby wybrać odpowiednie do utrwalenia załamania i kierować budową portali, zatrudniono Ciebie – nie zawiedź Bajtocji i jej imperatora! Podobno bardzo tego nie lubi...

## Komunikacja

**To jest zadanie interaktywne**. Twój program, zamiast czytać z wejścia dane i pisać odpowiedź na wyjście, powinien komunikować się z dostarczoną biblioteką. Za jej pomocą zdobędziesz informacje o kolejnych załamaniach czasoprzestrzennych, a także zadecydujesz o budowie kolejnych par portali. Aby użyć biblioteki, należy wpisać na początku programu:

- **C/C++**: `#include "cporlib.h"`

Biblioteka udostępnia następujące funkcje i procedury:

- `podzadanie()`;
  Ta funkcja zwraca numer podzadania, czyli liczbę całkowitą $T$ spełniającą $0 \leq T \leq 25$.
- `dajN()`, `dajM()`, `dajD()`;
  Pierwsza funkcja zwraca liczbę całkowitą $N$, która oznacza liczbę planet znajdujących się pod panowaniem Bajtazara.
  Druga funkcja zwraca liczbę całkowitą $M$, która oznacza liczbę załamań między planetami w Bajtocji.
  Trzecia funkcja zwraca liczbę całkowitą $D$, która oznacza ustalone przez Bajtazara ograniczenie górne na długość ścieżki między dwiema planetami.
  - **C/C++:**
    ```
    int dajN();
    int dajM();
    int dajD();
    ```
- `zalamanie()`;
  Funkcja zwraca pseudolosową parę różnych liczb całkowitych $a, b$ ($1 \leq a, b \leq N$) oznaczających numery planet, między którymi wystąpiło załamanie czasoprzestrzenne. Funkcję `zalamanie` możesz wywołać maksymalnie $M$ razy. Masz przy tym gwarancję, że kolejne wywołania funkcji `zalamanie` na tym samym teście zawsze zwrócą te same rezultaty.
  - **C/C++:** `pair<int, int> zalamanie();`



*Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022”.*

- `portal();`
  Funkcja `portal` buduje nową parę portali. Wybudowana para łączy te dwie planety, których numery zostały zwrócone przez ostanie wywołanie funkcji `zalamanie`. W szczególności Twój program nie może wywoływać funkcji `portal` przed pierwszym wywołaniem funkcji `zalamanie`.
  - **C/C++:** `void portal();`

Twój program musi wywołać funkcję `portal` **dokładnie** $N-1$ **razy**. Twój program **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.

# Podzadania

W każdym podzadaniu zachodzi $10 \leq N \leq 200\,000$, $10 \cdot N \leq M \leq 200\,000\,000$ oraz $log_2(N) \leq D \leq 100$. Każde podzadanie jest warte 4 punkty oraz zawiera 3 testy o wartościach $N, M, D$ podanych w tabeli.

| podzadanie | N | M | D |
|---|---|---|---|
| 0 (test przykładowy) | 10 | 1000 | 6 |
| 1 | 10 | 1 000 | 6 |
| 2 | 100 | 5 000 | 14 |
| 3 | 100 | 1 000 | 24 |
| 4 | 1 000 | 15 000 | 100 |
| 5 | 1 000 | 20 000 | 50 |
| 6 | 1 000 | 50 000 | 22 |
| 7 | 10 000 | 1 000 000 | 100 |
| 8 | 10 000 | 150 000 | 100 |
| 9 | 10 000 | 3 000 000 | 60 |
| 10 | 10 000 | 600 000 | 60 |
| 11 | 10 000 | 8 000 000 | 34 |
| 12 | 10 000 | 500 000 | 34 |
| 13 | 10 000 | 10 000 000 | 22 |
| 14 | 10 000 | 2 500 000 | 22 |
| 15 | 100 000 | 10 000 000 | 100 |
| 16 | 100 000 | 2 000 000 | 100 |
| 17 | 100 000 | 20 000 000 | 70 |
| 18 | 100 000 | 3 000 000 | 70 |
| 19 | 100 000 | 50 000 000 | 46 |
| 20 | 100 000 | 5 000 000 | 46 |
| 21 | 150 000 | 200 000 000 | 34 |
| 22 | 150 000 | 10 000 000 | 34 |
| 23 | 200 000 | 200 000 000 | 26 |
| 24 | 200 000 | 10 000 000 | 26 |
| 25 | 200 000 | 10 000 000 | 18 |

# Eksperymenty

W zakładce *Pliki* dostępna jest przykładowa biblioteka, która pozwoli Ci przetestować poprawność rozwiązania. Biblioteka zakłada, że $N = 10$, $M = 1\,000$ oraz $D = 6$. Biblioteka implementuje metody podane w zadaniu, jednak ich działanie różni się od tych opisanych w treści. **Należy samemu zaimplementować poprawne metody w bibliotece.** Po zakończeniu działania Twojego programu biblioteka wypisuje na standardowe wyjście informację o udzielonej odpowiedzi - jeżeli różni się ona od napisu *OK*, to działanie Twojego programu nie jest poprawne.

Do kompilacji rozwiązania wraz z biblioteką powinieneś użyć polecenia:

- **C++**: `g++ cporlib.cpp Twoj_plik.cpp -o Twoj_plik_wynikowy --std=c++11`

Plik z rozwiązaniem i biblioteka powinny znajdować się w tym samym katalogu.



*Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022”.*