Sygnalizacja chaosu

Limit pamięci: 256 MB, Limit czasu: 3s

Bajtek i Bajtusiątko dziękują ci za pomoc w zadaniu *Permutacja macierzy bez permutacji (i bez macierzy)*. Ponieważ sygnalizacja świetlna w Bajtocji mocno szwankowała, mieli mnóstwo czasu, aby nauczyć się, jak kodować dane na macierzy za pomocą cyfr.

Bajtek, jak na syna informatyka przystało, włamał się do sieci sygnalizacji świetlnej. „Skoro światła i tak nie działają, to dlaczego nie zagrać na nich w grę?" – pomyślał. Problem w tym, że zepsuta sygnalizacja potrafi spłatać figla...

Zasady gry:

1. Bajtek otrzymuje liczbę h.
2. Koduje ją w ustalony przez ciebie sposób: używając cyfr od 0 do 9, wypisuje macierz 10×10 (liczby oddzielone spacjami).
3. System (*chaos*) modyfikuje macierz: w każdym wierszu może dowolnie **przestawić** jej elementy, a następnie **kasuje** (zamienia na -1) od 0 do 2 liczb w całej macierzy.
4. Tak zaburzoną macierz otrzymuje Bajtusiątko, którego zadaniem jest ją odkodować i wypisać liczbę h, którą miał Bajtek.

Bracia wiedzą, że muszą współpracować, aby wygrać. Twoim zadaniem jest napisać program, który – uruchomiony raz jako Bajtek, a raz jako Bajtusiątko – pozwoli wygrać tę grę.

Interakcja

To zadanie jest interaktywne. Twój program zostanie uruchomiony w dwóch kopiach – jednej dla Bajtka i jednej dla Bajtusiątka. Każda kopia w pierwszym wierszu wejścia otrzyma słowo `Bajtek` albo `Bajtusiatko`, określające rolę tej kopii. Obie kopie w drugim wierszu otrzymają liczbę n – liczbę przypadków testowych.

W każdym przypadku testowym:

- kopia `Bajtek` otrzymuje w osobnym wierszu liczbę h i powinna wypisać zakodowaną macierz 10×10 (liczby oddzielone spacjami, po jednym wierszu macierzy w wierszu wyjścia);
- kopia `Bajtusiatko` otrzymuje zaburzoną macierz 10×10 i powinna wypisać jedną liczbę – odkodowane h.

Po wypisaniu każdego fragmentu należy opróżnić bufor wyjścia, np. poprzez `cout.flush()` lub `fflush(stdout)`.

W celu sprawdzenia poprawności formatu rozwiązania, możesz użyć symulatora chaosu (generatora permutacji i uszkodzeń) z pliku `sygchaos.cpp` dołączonego do zadania.

Wejście

Pierwszy wiersz wejścia zawiera słowo określające rolę kopii (`Bajtek` lub `Bajtusiatko`). Drugi wiersz zawiera liczbę przypadków testowych n.

Dalsza komunikacja przebiega zgodnie z opisem w sekcji *Interakcja*. W każdym przypadku testowym kodowana liczba spełnia $0 < h < 10^{20}$.

Wyjście

W odpowiedzi na każdy przypadek testowy program Bajtka wypisuje macierz 10×10, a program Bajtusiątka – jedną liczbę całkowitą równą odkodowanemu h.

Sygnalizacja chaosu

Limit pamięci: 256 MB, Limit czasu: 3s

Ograniczenia

$0 < h < 10^{20}$. Poszczególne podzadania spełniają dodatkowo:

Podzadanie	Ograniczenia	Punkty
1	$h < 10^4$	2
2	$h < 10^9$	12
3	$n < 10$	30
4	$h < 10^{12}$	26
5	$h < 10^{18}$	20
6	brak dodatkowych ograniczeń	10

Przykłady

Poniższy przykład zawiera trzy przypadki testowe ($h = 42$, $h = 10^{18}$ oraz $h = 7$). Poprawne rozwiązanie odtwarza dokładnie te liczby niezależnie od działania chaosu.

Wejście dla testu `syg0`:

```
3
42
1000000000000000000
7
```

Wyjście dla testu `syg0`:

```
42
1000000000000000000
7
```