



Задача Король щурів



*Nous vivrons. Et nous guérirons.
Les cicatrices... Nous les gardons.
Pour ne pas oublier. - Amicia*

Після запеклої битви з щурами, Аміції та Гуго час тікати від армії графа Віктора де Арля. Солдати розташувалися на вузькій стежці, яку можна представити як матрицю розміром $2 \cdot n$. Також відомо, що всього на цій стежці перебуває k солдатів.

Задача Аміція та Гюго визначають небезпеку конфігурації як кількість груп солдатів у ній. Більш формально, розглянемо бінарну матрицю розміром $2 \times n$, у якій на позиціях, де є солдат, записано 1. Будемо вважати, що дві клітинки з'єднані, якщо в обох із них записано 1 і вони мають спільну сторону. Зверніть увагу, що це відношення є транзитивним: це означає, що якщо клітинки a та b з'єднані, а клітинки b та c з'єднані, то a та c також вважаються з'єднаними. Зв'язаний компонент — це максимальна підмножина з'єднаних клітинок зі значенням 1. Рівень небезпеки — це кількість зв'язних компонентів в цій матриці.

Ваше завдання — допомогти двом героям знайти значення рівня небезпеки, розглядаючи всі можливі конфігурації. Кожна конфігурація вважається рівноймовірною. У цьому випадку очікуване значення можна визначити як середню кількість зв'язних компонентів для всіх можливих конфігурацій.

Деталі реалізації Вам потрібно реалізувати наступні функції:

```
void prec(int subtask_id)
```

```
int solve(int n, int k)
```

Перша функція буде викликана один раз на початку роботи градера. Ви можете використовувати її для попередньої обробки.

Друга функція має повернути очікуване значення небезпеки для заданих параметрів n та k за модулем 998 244 353. Формально, нехай $M = 998\,244\,353$. Можна показати, що відповідь може бути подана у вигляді нескоротного дробу $\frac{p}{q}$, де p та q — цілі числа, причому $q \not\equiv 0 \pmod{M}$. Поверніть ціле число, що дорівнює $p \cdot q^{-1} \pmod{M}$. Іншими словами, поверніть таке ціле число x , що $0 \leq x < M$ та $x \cdot q \equiv p \pmod{M}$.



Друга функція буде викликана t разів. Це означає, що у вхідних даних є кілька тестових наборів!

Не забудьте підключити заголовний файл "kog.h", інакше ви отримаєте помилку компіляції!

- **Обмеження**
- ◆ $1 \leq t \leq 10$
 - ◆ $1 \leq n \leq 10^9$
 - ◆ $0 \leq k \leq 10^6$
 - ◆ $k \leq 2 \cdot n$

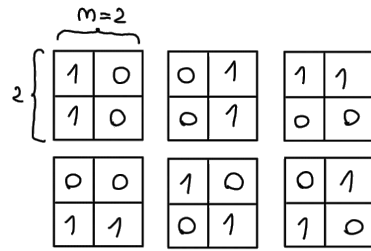
#	Бали	Обмеження
1	10	$1 \leq n \leq 100$
2	5	$1 \leq n \leq 2000$
3	5	$k \leq 3$
4	15	$k \leq 40$
5	10	$k \leq 400$
6	15	$k \leq 2000$
7	40	Ніяких обмежень

■ Приклади

Вхідні дані	Вихідні дані	Пояснення
2	332748119	Зверніть увагу, що градеру слід надати підзадачу тесту, кількість тестових наборів t та значення n, k для кожного тестового набору.
6	1	
2 2	518205646	
5 10	742082393	
2000 3	368118258	
2000 5	937239298	
100 32		
150 278		
7	0	
8	1	
1000000000 0	268791198	
1000000000 1	806373591	
1000000000 2	782159797	
1000000000 3	435727907	
5219873 192	712321002	
853875838 238	257644694	
43782384 1500		
58123292 180000		



Для першого тестового набору першого прикладу можливої конфігурації наведено нижче:



Всього є 6 конфігурацій, 4 з яких мають одну компоненту.

Отже, відповідь дорівнює $\frac{4 \cdot 1 + 2 \cdot 2}{6} = \frac{4}{3}$