

Завдання 3. Кролик

Божевільний Капелюшник щойно втратив свого улюбленого кролика (звісно, Білого Кролика) десь у послідовності з N клітинок і намагається його знайти. Клітинки пронумеровані цілими числами від 1 до N . Спочатку кролик знаходиться в одній невідомій клітинці послідовності, і кожна секунда пошуку Капелюшника проходить так:

1. Спочатку він вибирає одну клітинку послідовності та перевіряє її. Ми назвемо цю клітинку **перевіреною**. Якщо кролик є, пошук закінчується.
2. Після цього кролик вирішує або залишитися в тій самій клітинці, або перескочити до сусідньої (тобто на одну клітинку ліворуч або на одну клітинку вправо). **Зверніть увагу, що кролик може заскочити в перевірену клітинку, якщо вона сусідня; на цьому пошук не закінчується!**

Рішення кролика детерміновані з огляду на його настрій. Зокрема, у кролика є такі два настрої:

1. **Переляканий** настрій – коли кролик в такому настрої, він віддаляється далі від **перевіреної** клітинки. Якщо він не може відійти далі (тобто знаходиться в клітинці 1 або N), він залишається в тій же клітинці.
2. **Цікавий** настрій – коли кролик в такому настрої, він наближається до **перевіреної клітинки**. Зверніть увагу, що кролик завжди може підійти ближче.

Зверніть увагу, що кролик діє лише відповідно до останньої перевіреної клітинки і не піклується про попередні перевірені клітинки.

Оскільки це улюблений кролик Капелюшника, він дуже добре знає його настрій. Зокрема, він знає, що кролик чергує рівно S секунд **переляканого настрою** і C секунд **цікавого настрою**. Наприклад, якщо $S=2$ і $C=1$, то настрій кролика буде [переляканий, переляканий, цікавий, переляканий, переляканий, цікавий...].

Капелюшник дуже переживає за свого кролика і просить вас написати програму **rabbit.cpp**, яка обчислює послідовність комірок для перевірки, щоб кролик, незалежно від його початкового положення, був гарантовано знайдений.

Вхідні дані. Перший рядок вхідних даних містить три цілих числа: N , S і C , які описують кількість клітинок та поведінку кролика.

Вихідні дані. У першому рядку стандартного потоку ваша програма повинна надрукувати K , кількість секунд, яку займає ваша послідовність пошуку. У другому рядку ваша програма повинна надрукувати K цілих чисел у діапазоні $[1, N]$ з переліком клітинок, які перевіряються щосекунди. Зауважте, що ця послідовність може мати повтори.

Оцінювання. Нехай K — кількість секунд у вашій послідовності пошуку. Якщо ви спробуєте перевірити недійсну клітинку (тобто за межами діапазону $[1, N]$) або якщо ваша послідовність перевірки не завжди знаходить кролика, ви отримаєте 0 балів за тестовий приклад і вердикт «Wrong Answer».

В іншому випадку, якщо за тест дається T балів, ви отримаєте бали pT , де:

- $p = 0$, if $K > N$
- $p = 1$, if $K \leq \frac{N(S+L)}{S+2 \max(S, L)} + 3 \max(S, L)$
- $p = 0.3 \left(\frac{T}{K}\right)^2$, в інших випадках, де: $T = \frac{N(S+C)}{S+2 \max(S, C)} + 3 \max(S, C)$

Обмеження

$$2 \leq N \leq 10^4$$

$$0 \leq S, C \leq 50$$

Інформація про тестування

- Для 8% тестів $S = 1, C = 0$
- Для 12% тестів $S = 0, C = 1$
- Для 8% тестів $S = 1, C = 1$

Приклад

Введення	Виведення
12 2 1	14 2 5 3 2 6 1 2 11 12 12 8 10 12 6

Пояснення до прикладу

Можна перевірити, що незалежно від початкової позиції ця послідовність завжди знаходить кролика. Наприклад, розглянемо випадок, коли кролик починає з клітинки 8. Пошук буде відбуватися наступним чином:

Секунда	Перевірена клітинка	Настрій кролика (перед переходом)	Рух кролика
1	2	Переляканий	8 -> 9
2	5	Переляканий	9 -> 10
3	3	Цікавий	10 -> 9
4	2	Переляканий	10 -> 11
5	6	Переляканий	11 -> 12
6	1	Цікавий	12 -> 11
7	2	Переляканий	11 -> 12
8	11	Переляканий	12 -> 12
9	12	Цікавий	Знайдений

Для цього рішення маємо $K > T$, оскільки $K=14$ і $T=12$.

Таким чином, частина балів, отриманих за цей тест, становить $p \approx 0,26$.