

Task 3. Rabbit

Безумный Шляпник потерял своего любимого кролика (конечно же, это Белый Кролик) где-то в ряду из N последовательно расположенных ячеек и пытается найти его. Ячейки пронумерованы целыми числами от 1 до N . Исходно кролик находится в некоторой неизвестной ячейке последовательности, каждую секунду Безумный Шляпник делает следующее:

1. Сначала он выбирает одну ячейку и проверяет её. Будем называть такую ячейку **только что проверенной ячейкой**. Если кролик находится в этой ячейке, поиск завершается.
2. В противном случае кролик решает, либо он остаётся в ячейке, в которой находится, либо перепрыгивает в соседнюю ячейку (то есть на одну ячейку налево или на одну ячейку направо). **Заметьте, что кролик может перепрыгнуть в только что проверенную ячейку, и в этом случае поиск не завершается.**

Решения кролика однозначно определяются его настроением:

1. **Испуганный** кролик всегда перемещается **дальше от только что проверенной ячейки**. Если он не может переместиться дальше (то есть находится в ячейке 1 или N), он остаётся в текущей ячейке.
2. **Любопытный** кролик всегда перемещается **ближе к только что проверенной ячейке**. Обратите внимание, что кролик всегда может переместиться ближе.

Также обратите внимание, что кролик всегда ориентируется на только что проверенную ячейку и не помнит, какие ячейки были проверены раньше.

Безумный Шляпник хорошо изучил своего кролика, поэтому он хорошо знает его настроение. А именно, он знает, что настроение кролика меняется следующим образом: сначала он ровно S секунд является *испуганным*, затем оно ровно C секунд является *любопытным*, после чего настроение кролика повторяется. Например, если $S = 2$ а $C = 1$, настроение кролика задается последовательностью: *испуганный, испуганный, любопытный, испуганный, испуганный, любопытный, ...*

Шляпник просит вас написать программу `rabbit.cpp`, которая найдет последовательность ячеек, которые необходимо проверить, чтобы найти кролика, где бы он исходно не находился.

Input

На первой строке ввода находятся три целых числа: N , S и C , задающих число ячеек и параметры смены настроения кролика.

Output

На первой строке выведите K , количество секунд, которая занимает ваша последовательность проверок. На второй строке выведите K целых чисел в диапазоне $[1, N]$, ячейки, которые необходимо проверить. При необходимости вы можете проверять одну и ту же ячейку несколько раз.

Scoring

Пусть K – количество секунд в вашей последовательности проверок. Если вы пытаетесь проверить некорректную ячейку (за пределами диапазона $[1, N]$), или если ваша последовательность проверок **не всегда** находит кролика, вы получите 0 баллов за тест и результат проверки *Wrong Answer*.

Иначе, если тест стоит T баллов, вы получите за тест pT баллов, где:

- $p = 0$, если $K > 2N$
- $p = 1$, если $K \leq T$
- $p = 0.3 \left(\frac{T}{K}\right)^2$, иначе

Здесь: $T = \frac{N(S+C)}{S+2 \max(S,C)} + 3 \max(S, C)$

Constraints

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

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

Test Information

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

Sample test

Input	Output
12 2 1	14
	2 5 3 2 6 1 2 11 12 12 8 10 12 6

Sample test explanation

Вне зависимости от стартовой позиции приведенная последовательность всегда приведет к тому, что Шляпник найдет кролика. Например, пусть кролик исходно находился в ячейке 8. Тогда поиски будут происходить следующим образом:

Секунда	Только что проверенная ячейка	Настроение кролика (перед перемещением)	Перемещение кролика
1	2	Испуганный	8 -> 9
2	5	Испуганный	9 -> 10
3	3	Любопытный	10 -> 9
4	2	Испуганный	10 -> 11
5	6	Испуганный	11 -> 12
6	1	Любопытный	12 -> 11
7	2	Испуганный	11 -> 12
8	11	Испуганный	12 -> 12
9	12	Любопытный	Найден!

В этом решении $K > T$, так как $K = 14$, а $T = 12$.

Поэтому доля баллов, которое вы получите за этот тест, равно $p \approx 0.22$.