D. Guessing Game

Problem Name	Guessing Game
Time Limit	4 seconds
Memory Limit	1 gigabyte

В старом городе Лунд есть улица, на которой в ряд стоят N домов, пронумерованных от 0 до $N-1$. Эмма живет в одном из этих домов, а ее друзья Анна и Бертиль хотят выяснить, в каком именно. Вместо того чтобы просто рассказать друзьям, где она живет, Эмма решила поиграть с ними в игру. Перед началом игры Анна и Бертиль знают только количество домов на улице. В этот момент Анна и Бертиль могут выбрать положительное целое число K и договориться о стратегии. После этого любое общение запрещено.

Сама игра состоит из двух этапов. На первом этапе Эмма выбирает порядок посещения домов таким образом, чтобы ее дом был последним. Затем она ведет Анну к домам по очереди в этом порядке, не сообщая порядок Анне заранее. В каждом доме, который не является домом Эммы, Анне разрешается написать на входной двери дома мелом одно целое число от 1 до K. В последнем доме, который они посещают (в доме Эммы), Эмма сама пишет на двери целое число от 1 до K.

Во второй фазе игры Бертиль проходит по улице от дома 0 до дома $N-1$ и читает все номера, написанные на дверях Анной и Эммой. Теперь он хочет угадать, в каком доме живет Эмма. У него есть два шанса угадать правильный ответ, и если ему это удастся, то он и Анна выиграют игру. В противном случае в игре побеждает Эмма.

Сможете ли вы разработать стратегию, при которой Анна и Бертиль гарантированно выиграют игру? Ваша стратегия будет оцениваться по значению K (чем меньше, тем лучше).

Implementation

Ваше решение будет запущено несколько раз на одном и том же тесте. При первом запуске будет выполнена стратегия Анны. После этого будет выполнена стратегия Бертиля.

Первая строка ввода содержит два целых числа — P и N, где P — либо 1, либо 2 (номер фаза), а N — количество домов. **За исключением примера из условия задачи (не используется для подсчета баллов), N всегда будет равно $100\,000$.**

Следующие входные данные зависят от фазы:

Фаза 1

Ваша программа должна начать с вывода в единственной строке числа K $(1 \leq K \leq 1\,000\,000)$. Затем, $N-1$ раз она должна прочитать строку, содержащую индекс i $(0 \leq i < N)$, и вывести строку с целым числом A_i $(1 \leq A_i \leq K)$, где A_i — это число, которое Анна написала на двери дома i. Каждый индекс i, кроме индекса дома Эммы, будет встречаться ровно один раз, в некотором порядке, определяемом тестирующей программой.

Фаза 2

Ваша программа должна прочитать строку с N целыми числами, $A_0, A_1, \ldots, A_{N-1}$, где A_i — номер, написанный на двери дома i.

Затем следует вывести строку с двумя целыми числами s_1 и s_2 $(0 \leq s_i < N)$, угаданные индексы. Допускается, что s_1 и s_2 равны.

Implementation Details

Обратите внимание, что когда вы выполняете программу на Фазе 2, происходит перезапуск программы. Это означает, что вы не сможете сохранить информацию в некоторых переменных между запусками.

После каждой выведенной строки не забудьте сделать flush, иначе программа может быть оценена как Time Limit Exceeded. В Python функция `print()` делает flush автоматически. В C++ `cout << endl;` помимо вывода новой строки, также делает flush. Если используется printf, то используйте `fflush(stdout)`.

Тестирующая программа для этой задачи может быть **адаптивной**, т.е. она может менять свое поведение в зависимости от результатов работы вашей программы, чтобы предотвратить прохождение эвристических решений. Она может выполнить пробный запуск фазы 1, посмотреть на ваш результат, а затем запустить фазу 1 по-настоящему, используя информацию, полученную в результате предыдущего запуска.

Ваша программа *должна* быть детерминированной, т.е. вести себя одинаково, если ее дважды выполнить на одних и тех же входных данных. Если вы хотите использовать рандом в своей программе, то обязательно используйте фиксированный random seed, чтобы ваше решение было детерменированным. Это можно сделать, передав константу в `srand` (в C++) или `random.seed` (в Python). Если тестирующая программа обнаружит, что ваша программа не является детерминированной, то вы получите вердикт Wrond Answer.

Если *сумма* времени выполнения (до 3) отдельных запусков вашей программы превысит лимит времени, то ваша отправка будет оценена как Time Limit Exceeded.

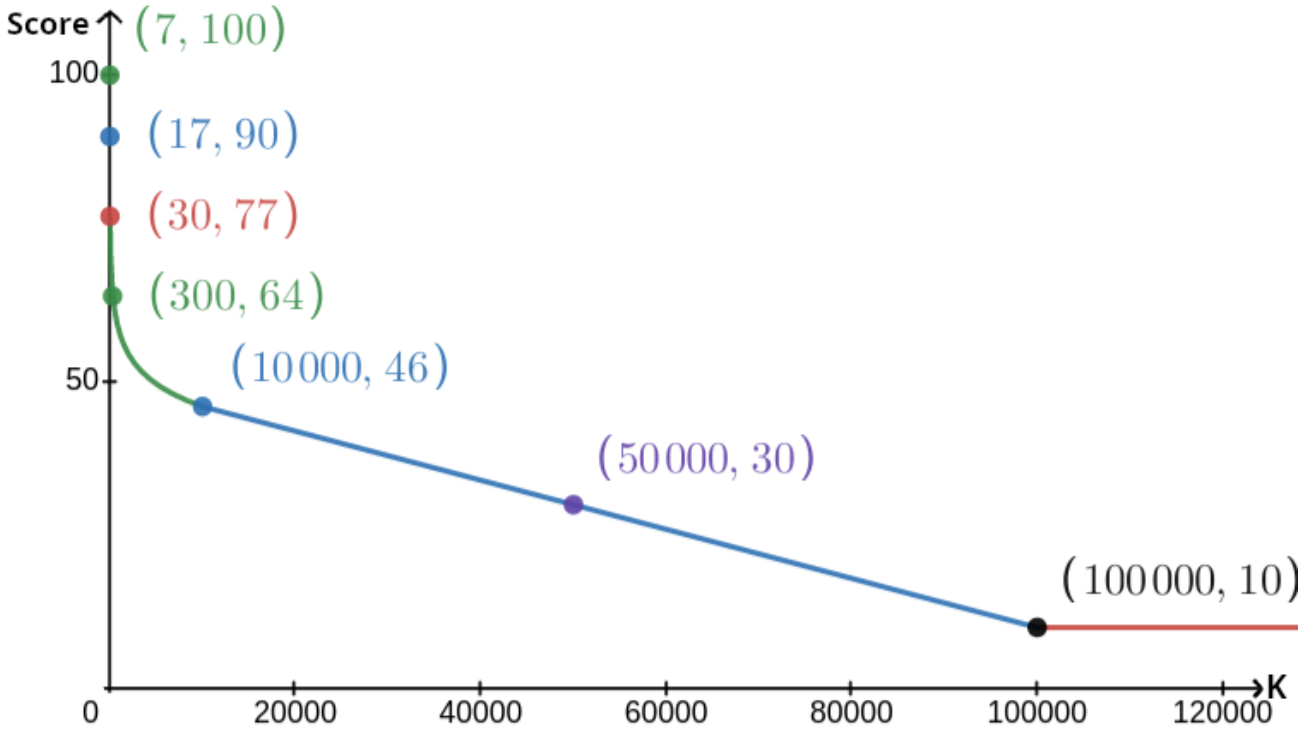
Scoring

Ваше решение будет протестировано на ряде тестов. Если ваше решение не справится с *любым* из этих тестов (например, даст неправильный ответ (Wrong Answer), аварийно завершит работу (Run-Time Error), превысит лимит времени (Time Limit Exceeded) и т.д.), вы получите 0 баллов и соответствующий вердикт.

Если ваша программа нашла индекс дома Эммы во *всех* тестах, то вы получите вердикт Accepted и оценку, которая рассчитывается следующим образом. Пусть K_{max} - максимальное значение K, используемое для любого теста. В зависимости от K_{max}:

	Score
$K_{max} > 99\,998$	10 баллов
$10\,000 < K_{max} \leq 99\,998$	$10 + \lfloor 40(1 - K_{max}/10^5) \rfloor$ баллов
$30 < K_{max} \leq 10\,000$	$46 + \lfloor 31(4 - \log_{10}(K))_{max}))/(4 - \log_{10}(30)) \rfloor$ баллов
$7 < K_{max} \leq 30$	$107 - K_{max}$ баллов
$K_{max} \leq 7$	100 баллов

Функция оценки изображена на рисунке ниже.



Пример из условия задачи не учитывается при подсчете баллов, и ваше решение не должно на нем работать.

Testing Tool

Чтобы облегчить тестирование вашего решения, мы предлагаем простой инструмент, который вы можете скачать. См. раздел "Приложения" в нижней части Kattis страницы задачи. `testing_tool` является необязательным для использования, и вы можете вносить в него изменения. Обратите внимание, что официальная тестирующая программа на Kattis отличается от `testing_tool`.

Пример использования (со значениями $N = 4$, $s = 2$, где s — номер, написанный на последнем посещенном доме):

Для программ на языке Python — `solution.py` (обычно запускается как `pypy3 solution.py`):

```
python3 testing_tool.py pypy3 solution.py <<<"4 2"
```

Для программ на C++, сначала скомпилируйте ее (например с помощью `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) и после запустите:

```
python3 testing_tool.py ./solution.out <<<"4 2"
```

`testing_tool` будет посещать дома в случайном порядке. Чтобы использовать фиксированный порядок, измените инструмент тестирования там, где написано "MODIFY HERE".

Example Interaction

Пример из условия задачи не учитывается при подсчете баллов, и ваше решение не должно на нем работать.

Предположим, что у нас $N = 4$ и что Эмма живет в доме 1. Пусть A — список чисел, записанных на домах. Первоначально $A = [0, 0, 0, 0, 0]$, где 0 означает, что на соответствующем доме не написано ни одного числа.

В первом запуске вашей программы:

$N = 4$. Ваше решение выводит с $K = 3$.

Для A_2, ваше решение выводит 3. Теперь A — это $[0, 0, 3, 0]$.

Для A_0, ваше решение выводит 1. Теперь A — это $[1, 0, 3, 0]$.

Для A_3, ваше решение выводит 2. Теперь A — это $[1, 0, 3, 2]$.

Наконец, для A_1, ваше решение выводит 2, так что в итоге $A = [1, 2, 3, 2]$. На этом заканчивается первая фаза.

В фазе 2 ваша программа считывает `1 2 3 2`.

И выводит `1 3`.

Поскольку одно из предположений является правильным индексом дома (1), Анна и Бертиль выигрывают игру.

grader output	your output
1 4	
	3
2	
	3
0	
	1
3	
	2

grader output	your output
2 4	
1 2 3 2	
	1 3