

D. Игра "Познай"

Име	Игра "Познай"
Time Limit	4 seconds
Memory Limit	1 gigabyte

В стария град на Лунд има улица с N къщи, наредени в редица и номерирани от 0 до $N - 1$. Ема живее в една от тези къщи и нейните приятели Ана и Бертил искат да разберат кой е номера на нейната къща. Вместо просто да каже на приятелите си къде живее, Ема решава да играят игра.

Преди да започне играта, Ана и Бертил знаят само броя на къщите по улицата. В този момент Ана и Бертил могат да изберат положително цяло число K и да се споразумеят за стратегия. Всякаква комуникация между тях след това е забранена.

Играта се състои от две фази. В първата фаза Ема избира подредба за посещаване на къщите, в която нейната къща е последната. След това тя води Ана през къщите една по една в този ред, без да ѝ казва предварително реда. За всяка къща, която не е нейната, Ана има право да напише едно произволно цяло число между 1 и K на вратата с пастел. За последната къща, която посещават, а именно къщата на Ема, Ема сама записва цяло число между 1 и K на вратата.

Във втората фаза на играта, Бертил се разхожда по улицата от къща с номер 0 до тази с номер $N - 1$ и чете всички числа, написани на вратите от Ана и Ема. Сега той иска да познае в коя къща живее Ема. Той има две възможности да познае правилно, и ако успее, той и Ана печелят играта. В противен случай, печели Ема.

Можете ли да измислите стратегия, с която Ана и Бертил гарантирано да спечелят играта?

Вашата стратегия ще бъде оценявана на база стойността на K (колкото е по-малка, толкова по-добре).

Имплементация

Вашата програма ще бъде изпълнявана няколко пъти. При първото изпълнение тя ще изпълни стратегията на Ана. След това, при ново стартиране на програмата ще се изпълни стратегията на Бертил.

Първият ред от входа ще съдържа две цели числа P и N , където P е или 1, или 2 (първа или втора фаза), а N е броят на къщите.

Освен за примерния вход (който не се използва за оценка), N ще е винаги равно на 100 000.

Следващият вход зависи от фазата на играта.

Фаза 1

Вашата програма трябва да започне, извеждайки числото K на един ред ($1 \leq K \leq 1\,000\,000$). След това, $N - 1$ пъти, трябва да прочете ред, съдържащ индекс i ($0 \leq i < N$) и да изведе ред с числото A_i , където A_i е числото, което Ана пише с пастел на къща номер i . Всеки номер i , с изключение на номера на къщата на Ема ще се срещне точно веднъж.

Фаза 2

Вашата програма трябва да прочете ред с N цели числа A_0, A_1, \dots, A_{N-1} , където A_i е числото написано с пастел на къща с номер i .

След това, трябва да изведе ред с две цели числа s_1 и s_2 ($0 \leq s_i < N$) - предположените индекси за къщата на Ема. s_1 и s_2 могат да са равни.

Детайли по имплементацията

Забележете, че когато се пуска фаза 2, вашата програма бива пусната отново. Това означава, че не можете да запазвате информация в никакви променливи между фазите.

След всяко извеждане на ред от изхода, уверете се, че изчиствате стандартния изход (flush), в противен случай програмата ви може да бъде оценена с "Time Limit Exceeded" (превишено време за изпълнение). В Python, `print()` автоматично изчиства стандартния изход. В C++, `cout << endl;` също изчиства стандартния изход, освен да отпечата нов ред; ако използвате `printf`, използвайте `fflush(stdout)`.

Оценяващата система за тази задача може да бъде **адаптивна**, което означава, че може да промени своето поведение в зависимост от изхода на вашата програма, за да предотврати преминаването на евристични решения. Възможно е да изпълни първата фаза, да разгледа вашия изход и след това да изпълни отново първата фаза, използвайки информация, извлечена от първото изпълнение.

Вашата програма трябва да бъде детерминистична, т.е. да се държи по същия начин, ако се изпълни два пъти със същите входни данни. Ако искате да използвате случайност в програмата си, уверете се, че използвате фиксиран *random seed*. Това може да се направи, като подадете вградена константа на `srand` (в C++) или `random.seed` (в Python). В частност,

вашата програма не може да използва `srand(time(NULL))` в C++. Ако оценяващата система засече, че вашата програма не е детерминистична, ще получите "Wrong Answer" (грешен отговор).

Ако сумата от времената на изпълнение на (до 3) отделни пускания на вашата програма превиши времеви лимит, вашето решение ще бъде оценено с "Time Limit Exceeded" (превишен времеви лимит).

Оценяване

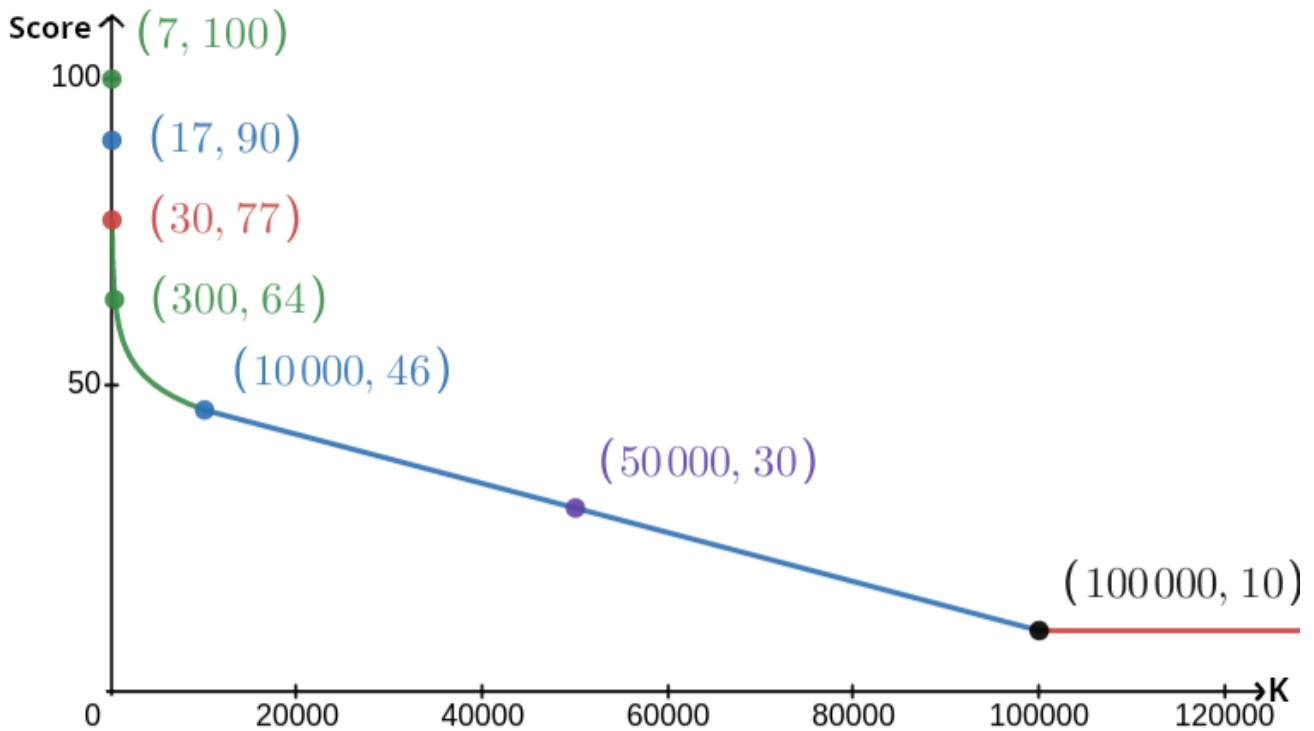
Вашето решение ще бъде тествано върху няколко тестови случаи. Ако вашето решение не успее в *някой* от тези случаи (например, като даде неправилен отговор (Wrong Answer), се срина (Run-Time Error), превиши времеви лимит (Time Limit Exceeded) и други), ще получите 0 точки и съответното съобщение за грешка.

Ако вашата програма успешно открие тайния номер на къщата на Ема във всички тестови случаи, ще получите съобщение "Accepted" и оценка, изчислена по следния начин:

Нека K_{max} е максималната стойност на K , използвана в *някой* от тестовите случаи. В зависимост от K_{max} :

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

Броят точки може да видите и от схемата по-долу:



Тестовият случай ще бъде игонириран по време на оценяването и вашето решение не е задължително да дава верен отговор на него.

Локално тестване

За улеснение при тестване на вашето решение, Ви е предоставен прост инструмент, който можете да изтеглите. Вижте "прикачените файлове" в долната част на страницата на задачата в Kattis. Този инструмент не е задължителен за използване и имате право да го промените. Обърнете внимание, че официалният програмен оценяващ инструмент на Kattis е различен от предоставения тестов инструмент.

Примерно използване (с $N = 4$, $s = 2$, където s е числото, което ще бъде написано на последната къща (къщата на Ема)):

За програми на Python, с име "solution.py" (нормално изпълнявани като "python3 solution.py"):

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

За програми на C++, първо ги компилирайте (например с `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) след това ги стартирайте:

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

Тестовият инструмент ще посещава къщите в случаен ред. За да използвате специфичен ред, променете тестовия инструмент на мястото, където пише "MODIFY HERE".

Примерно взаимодействие

Тестовият случай ще бъде игонириран по време на оценяването и вашето решение не е задължително да дава верен отговор на него.

Нека $N = 4$ и Ема живее в къща 1. Нека A е списъкът с числа, написани на къщите. Първоначално, $A = [0, 0, 0, 0]$, където 0 означава, че няма написано число на съответната къща.

При първото изпълнение на вашата програма:

Дава се $N = 4$. Решението Ви отговаря с $K = 3$.

Пита се за A_2 . Решението Ви отговаря с 3. A става $[0, 0, 3, 0]$.

Пита се за A_0 . Решението Ви отговаря с 1. A става $[1, 0, 3, 0]$.

Пита се за A_3 . Решението Ви отговаря с 2. A става $[1, 0, 3, 2]$.

Накрая, оценяващото решение задава $A_1 = 2$, така че $A = [1, 2, 3, 2]$ в края. Това завършва първата фаза.

При второто изпълнение на вашата програма, се предава списъкът 1 2 3 2.

Решението отговаря с 1 3.

Тъй като едно от предположенията е правилният номер на къщата (1), Ана и Бертил печелят играта.

Изход на журито	Ваш изход
1 4	
	3
2	
	3
0	
	1
3	
	2

Изход на журито	Ваш изход
2 4	
1 2 3 2	
	1 3