

### 3. Нейтринный излучатель (neu)

1 сек

100 очков

Недавно астрономы узнали, что в звездной системе  $\zeta$ -2019A, состоящей из  $N$  планет, есть ровно одна обитаемая планета. Но эта система расположена очень далеко, поэтому они могут исследовать ее только с помощью нейтринного излучателя.

Прибор работает следующим образом. Сначала он посылает очень мощный поток частиц, который можно охарактеризовать двумя целыми числами  $X$  и  $Y$ . Затем ученые анализируют отраженные частицы и получают целое число  $B = \text{gcd}(X, A + Y)$ , где  $A$  — номер обитаемой планеты (планеты пронумерованы целыми числами от 1 до  $N$ , от самой ближней к звезде  $\zeta$ -2019A планеты до самой дальней), а  $\text{gcd}(X, Y)$  обозначает наибольший общий делитель чисел  $X$  и  $Y$ , то есть наибольшее целое число на которое как  $X$  так и  $Y$  делятся без остатка. Нейтринный излучатель — прибор энергозатратный, поэтому разрешается запустить поток частиц не более, чем 40 раз.

Теперь требуется программа для нейтринного излучателя, которая, посылая потоки частиц, сможет определить номер обитаемой планеты в звездной системе  $\zeta$ -2019A. Ядром этой программы является функция `int Locate(int n)`, которая будет вызвана ровно один раз, с аргументом  $N$ , означающим число планет в звездной системе. По окончании работы, функция должна вернуть целое число  $A$  ( $1 \leq A \leq N$ ) — номер обитаемой планеты.

Функция `Locate` может вызывать функцию `int Scan(int x, int y)`, которая запускает излучатель с параметрами  $X$  и  $Y$  ( $1 \leq X \leq 10^9$ ,  $0 \leq Y \leq 10^9$ ), а затем возвращает результат анализа отраженных частиц  $B = \text{gcd}(X, A + Y)$ . Функцию `Scan` можно вызвать не более, чем 40 раз.

В тестовой среде есть файлы-примеры, где нужные функции уже описаны и как решение нужно дописать реализацию функции `Locate`. Также в файл решения можно записывать и свои функции. Для тестирования своего решения на своём компьютере есть пример оценивающей программы, описание входных и выходных данных которой приведено ниже (на сервере используется другая оценивающая программа, которая проверяет также и верность ответов, возвращаемых решением). Для компиляции и тестирования своего решения на своём компьютере:

Язык	Решение	Команды
C++	neu.cpp	<pre>g++ -o grader grader.cpp neu.cpp ./grader</pre>
Java	neu.java	<pre>javac -cp . grader.java neu.java java -cp . grader</pre>
Python	neu.py	<pre>python grader.py</pre>

**Входные данные.** В единственной строке текстового файла `neusis.txt` два целых числа: количество планет  $N$  в звездной системе  $\zeta$ -2019A ( $1 \leq N \leq 10^9$ ) и номер  $A$  обитаемой планеты ( $1 \leq A \leq N$ ).

**Выходные данные.** В текстовый файл `neuval.txt` выводится лог общения оценивающей программы и функции `Locate`.

**Пример.**

<code>neusis.txt</code>	<code>neuval.txt</code>
5 3	Locate(5)
	Scan(3, 6) = 3
	Scan(4, 2) = 1
	Scan(2, 1) = 2
	Scan(4, 5) = 4
	result = 3

В звездной системе  $\zeta$ -2019A всего 5 планет и обитаемой является планета номер 3. Вызов `Scan(3, 6)` вернет  $\gcd(3, 3+6) = \gcd(3, 9) = 3$ ; `Scan(4, 2)` вернет  $\gcd(4, 3+2) = \gcd(4, 5) = 1$ ; `Scan(2, 1)` вернет  $\gcd(2, 3+1) = \gcd(2, 4) = 2$ ; `Scan(4, 5)` вернет  $\gcd(4, 3+5) = \gcd(4, 8) = 4$ . В конце функция `Locate` возвращает оценивающей программе значение 3.

**Оценивание.** В этом задании тесты поделены на группы. За каждую группу очки получают только те решения, которые пройдут все тесты этой группы. В группах выполняются следующие дополнительные условия:

1. (11 очков)  $N \leq 40$ ;
2. (8 очков)  $N \leq 80$ ;
3. (12 очков)  $N \leq 150$ ;
4. (9 очков)  $N \leq 10^3$ ;
5. (18 очков)  $N \leq 10^4$ ;
6. (23 очка)  $N \leq 2 \cdot 10^5$ ;
7. (19 очков) дополнительных условий нет.