Задача CT33. ТАЙНА

След като наредили всички гормити по местата им, Лазар и Яна решили да играят на друга игра. За съжаление не успели да намерят достатъчно хора за Скрабъл, Експлодиращи котета и Кодови имена. На Яна обаче ѝ хрумнала добра идея.

Тя измислила тайната аритметична операция $\star$. За две неотрицателни числа a и b, по-малки или равни на 10^9, съществува число, спазващо същото ограничение, и равно на $a \star b$. Операцията била асоциативна, тоест равенството $(x \star y) \star z = x \star (y \star z)$ е в сила за всеки три x, y и z, където $0 \leq x, y, z \leq 10^9$. Обърнете внимание, че е възможно $x \star y \neq y \star x$.

Яна избира N числа $A_0, A_1 \ldots A_{N-1}$. След това започва да задава въпроси на Лазар от вида: "Колко е стойността на $A_L \star A_{L+1} \star \ldots \star A_R$?". Като подсказка, Яна може да му сподели за кои да е две неотрицателни числа, по-малки или равни на 10^9 каква е получената стойност след прилагането на операцията $\star$ върху тях. Лазар може да пита Яна още когато получи числата, както и по време на нейните заявки.

Задача

Лазар знае, че сте топ програмист и би желал да му помогнете да намали броя на запитвания към малката сестричка.

Детайли по имплементацията

Решението ви трябва да съдържа функцията `Init` със следния прототип:

```
void Init(int N, int A[]);
```

Тя ще бъде извикана веднъж в началото. Предоставят ви се броят на числата N и самите стойности – $A_0, A_1, \ldots, A_{N-1}$. Също така, решението ви трябва да съдържа функцията `Query` със следния прототип:

```
int Query(int L, int R);
```
$(0 \leq L \leq R \leq N-1)$

Тя ще бъде извикана многократно и задължително след `Init`. Функцията трябва да връща отговора на Лазар – стойността на $A_L \star A_{L+1} \star \ldots \star A_R$. За комуникация с журито (Яна) е предоставена функцията `Secret`:

```
int Secret(int X, int Y);
```

За две цели неотрицателни числа X и Y, където $0 \leq X, Y \leq 10^9$, функцията връща стойността на $X \star Y$. *Ако подадените параметри не спазват ограничението за $\star$, ще получите съобщение "Wrong answer".*

Вашата програма трябва да имплементира функциите `Init` и `Query`, но не трябва да съдържа функцията `main`. Освен това, тя не трябва да чете на стандартния вход или да печата на стандартния изход. Програмата ви също така трябва да включва хедър файла `secret.h` чрез указание към предпроцесора `#include "secret.h"`. Стига да спазва тези условия, програмата ви може да съдържа каквито и да е помощни функции, променливи, константи и прочее.

Ограничения

$1 \leq N \leq 1000$

$0 \leq Ai \leq 1\,000\,000\,000$ ($0 \leq i \leq N - 1$)
Функцията `Query` ще бъде извикана максимално 10000 пъти.

Оценяване

Програмата ви ще бъде оценена при условие, че отговаря правилно на всички викания на Query и не получава съобщение *"Wrong answer"*. Резултатът ви ще бъде:

(1) 100 точки, ако следните условия са изпълнени:

- в `Init`, броят на извиквания на `Secret` е най-много 8000.
- за всяко извикване на `Query`, броят на допитванията до `Secret` е най-много 1.

(2) 30 точки, ако програмата ви не спазва (1) и следните условия са изпълнени:

- в `Init`, броят на извиквания на `Secret` е най-много 8000.
- за всяко извикване на `Query`, броят на допитванията до `Secret` е най-много 20.

(3) 6 точки, ако програмата не спазва (1) и (2)

Примерна комуникация

Вход
8 1 4 7 2 5 8 3 6 4 0 3 1 7 5 5 2 4

Извикване	**Върната стойност**
`Init(8, [1, 4, 7, 2, 5, 8, 3, 6])`	-
`Query(0, 3)`	13
`Query(1, 7)`	32
`Query(5, 5)`	8
`Query(2, 4)`	13

Процедурата `Secret` може да бъде извиквана по време на изпълнението на `Init`, както и на `Query`. Например при извикването `Secret`(4, 7), върнатата стойност е 10, тъй като $4 \star 7 = 10$, поради аритметичната операция, която примерният грейдър използва. Стойността на $1 \star 4 \star 7 \star 2$ се изисква в първата завка.

От $1 \star 4 \star 7 \star 2 = (1 \star (4 \star 7)) \star 2 = (1 \star 10) \star 2 = 11 \star 2 = 13$ при използване на примерния грейдър, `Query` следва да върне 13.

Локално тестване

В системата е предоставен файлът **`Lgrader.cpp`**, чрез който може да тествате локално програмата си. За целта трябва да добавите `#include "Lgrader.cpp"` към кода си.

От първия ред на стандартния вход на грейдъра се въвежда N. От втория ред следват стойностите на $A_0, A_1, \ldots, A_{N-1}$. Следва Q – броя на заявките. От следващите редове се въвеждат Q двойки числа – лявата и дясната граница от въпросите към Лазар.

Аритметичната операция $\star$ в `Lgrader.cpp` е дефинирана като:

$$x \star y = min\{x + 2\left\lfloor\frac{y}{2}\right\rfloor, 1\,000\,000\,000\}.$$

Обърнете внимание, че тази операция няма да съвпада с оригиналната операция в истинския грейдър.

Ако програмата ви връща верни резултати, грейдърът ще отпечатва отговорите за всяка заявка, както и броя на извиквания на процедурата `Secret` в `Init` и максималния брой в `Query`. В противен случай ще получите съобщение за *"Wrong answer"*.