

decrypt

100 точкиSource code: **decrypt.c, decrypt.cpp, decrypt.pas**Time limit: **0.1 seconds**Memory limit: **64MB**

Разнесе се слух, че Научният комитет използва специално устройство за криптиране на комуникациите си. Ако вие може да го кракнете, ще може да узнаете задачите предварително. Миналата нощ един от членовете на Научния комитет е забравил своето устройство в нощния бар. След като сте го намерили, вие сте го отворили и виждате принципната му схема:

Всички операции използват 8 бита. XOR е побитовата операция изключващо „или” (означена с \wedge в езика C, и с `xor` в езика Pascal).

R е последователност от псевдослучайни числа, дефинирана по следния начин:

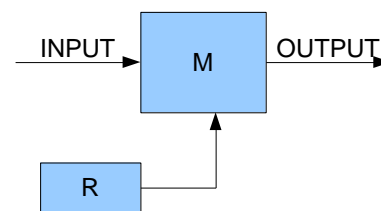
- R[0], R[1] и R[2] имат секретни стойности, известни само на Научния комитет.
- За $n = 3, 4, \dots$ $R[n] = R[n-2] \text{ XOR } R[n-3]$.

Освен това имаме функция M, $M: [0..255] \rightarrow [0..255]$, която е взаимно еднозначно съответствие, т.е. ако $x \neq y$, то $M[x] \neq M[y]$.

Устройството на Научния комитет приема по едно число за всеки момент време и го извежда като криптирано. След използване на устройството N пъти, следващото число INPUT се кодира така:

- $\text{OUTPUT} = M(\text{INPUT XOR } R[N])$

Даже и да разберете как работи устройството, вие не знаете стойностите R[0], R[1] и R[2]. Вие не знаете и как работи функцията M, така че не можете да декодирате. Това, което може да правите, е да си играете с устройството. Вие може да му задавате числа за вход на устройството и да виждате какво ще изведе то.



Задача

Вашата задача е да намерите стойностите: R[0], R[1], R[2], M[0], M[1], .., M[255] чрез използване на по-малко от **320** запитвания (входни числа).

Ограничения

- Правилно решение получава точки само ако броят на заявките е по-малък от 320.
- Във всички тестове, стойностите на търсените от вас числа R[0], R[1], R[2], M[0], M[1], ..., M[255] са случайни.
- $0 \leq \text{INPUT}, \text{OUTPUT}, R, M \leq 255$

Взаимодействие (Interaction)

Това е интерактивна програма. За да играете с модула, напишете число между 0 и 255 в стандартния изход. След това, прочетете от стандартния вход числото, изработено от устройството на Научния комитет, което също е цяло число между 0 и 255. Когато научите търсените стойности, изведете един ред, започващ с думата **SOLUTION** и след това изведете изход от 259 реда, съдържащи $R[0]$, $R[1]$, $R[2]$, $M[0]$, $M[1]$, ..., $M[255]$, по едно число на ред.

Инструкции за програмиране

След всеки ред, изведен на стандартния изход, програмистите на C трябва да използват `fflush(stdout)`, а програмистите на Pascal – `flush(output)`.

C	C++	Pascal
<code>printf("%d\n", q); fflush(stdout);</code>	<code>cout<<q<< '\n'; cout.flush();</code>	<code>writeln(q); flush(output);</code>

Пример

Нека секретните стойности са $R[0] = 0$, $R[1] = 1$, $R[2] = 3$ и $M[i] = (i + 1) \text{ MODULO } 256$. Оттук намираме, че $R[3] = 1$.

Contestant output	Contestant input	Обяснение
10	11	$M[10 \text{ XOR } 0] = M[10] = 11$
10	12	$M[10 \text{ XOR } 1] = M[11] = 12$
11	9	$M[11 \text{ XOR } 3] = M[8] = 9$
12	14	$M[12 \text{ XOR } 1] = M[13] = 14$
...		
SOLUTION 0 1 3 1 2 3 4 ... 254 255 0		Когато намерите секретните стойности, трябва да ги изведете.