

# Максимизировать сумму XOR

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Будем обозначать как  $\oplus$  операцию *побитового «исключающего или»* для целых чисел. В языках программирования C++ и Java она обозначается символом « $\wedge$ », в паскале и Python — ключевым словом «xor». Например,  $9 \oplus 3 = 1001_2 \oplus 11_2 = 1010_2 = 10$ .

Даны два массива  $A$  и  $B$  длины  $n$ . Обозначим как  $X(A)$  для массива  $A$  результат вычисления побитового «исключающего или» от всех элементов массива:  $X(A) = A_1 \oplus A_2 \oplus \dots \oplus A_n$ . Аналогично, введем обозначение  $X(B) = B_1 \oplus B_2 \oplus \dots \oplus B_n$ .

Для каждого  $i$  от 1 до  $n$  разрешается поменять местами элементы  $A_i$  и  $B_i$ . Необходимо определить, какие из этих обменов надо сделать, чтобы максимизировать сумму  $X(A) + X(B)$ .

## Формат входных данных

В первой строке входных данных находится число  $n$  — количество элементов ( $1 \leq n \leq 10^5$ ). В следующей строке находится  $n$  элементов массива  $A$  ( $0 \leq A_i \leq 10^{18}$ ). В следующей строке в таком же формате дан массив  $B$ .

## Формат выходных данных

В первой строке выведите максимальную возможную сумму и число  $k$  — количество необходимых обменов. В следующей строке выведите  $k$  различных чисел от 1 до  $n$  — индексы элементов, которые надо поменять.

## Пример

стандартный ввод	стандартный вывод
2	6 1
1 1	1
2 2	

## Замечание

В примере после обмена массивы равны  $A = [2, 1]$  и  $B = [1, 2]$ , соответственно.  
 $X(A) = 2 \oplus 1 = 10_2 \oplus 1_2 = 11_2 = 3$ ,  $X(B) = 3$ ,  $X(A) + X(B) = 6$ .