XORdenamiento

Se da una lista de n enteros de 0 a $2^{20} - 1$.

Se puede realizar la siguiente operación cualquier cantidad de veces (cero o más).

Elegir dos posiciones **distintas** $1 \leq x, y \leq n$ y asignar:

$$a_x := a_x \text{ XOR } a_y$$

Donde **XOR** es el XOR bit a bit de los dos números.

Hay que realizar estas operaciones para que la lista resultante esté ordenada de forma no decreciente (es decir, $a_i \leq a_{i+1}$ para cada posición i de 1 a $n - 1$).

¿Puedes construir una secuencia de operaciones suficientemente corta para solucionar esta tarea?

Entrada y salida

La primera línea contiene el número n, el tamaño de la lista.

La segunda línea contiene n enteros $a_1, \ldots, a_n$.

En la primera línea, imprime el número de operaciones m.

En las siguientes m líneas, imprime las posiciones x y y.

Ejemplo

Entrada:

```
5
0 1 1 1 2
```

Salida:

```
0
```

Entrada:

```
5
4 3 2 1 0
```

Salida:

```
5
5 1
1 5
2 3
2 4
4 3
```

Restricciones

$n = 1000$ en todos los casos.

Los ejemplos muestran casos con $n < 1000$ solamente para ayudarte. No aparecerán en los casos de prueba reales.

$0 \leq a_i \leq 2^{20} - 1$.

Puedes hacer como máximo 21000 operaciones.

Subtareas

1. (100 puntos) Sin restricciones adicionales.

Adicionalmente, la puntuación que obtienes depende del número de operaciones que hagas: para obtener una puntuación completa debes realizar como máximo 2500 operaciones y para obtener una puntuación positiva debes realizar como máximo 21000 operaciones. La puntuación de cada subtarea es multiplicada por un multiplicador $M(m)$, donde m es el máximo número de operaciones que has hecho en los casos de esa subtarea. El valor de $M(m)$ viene dado por:

$$M(m) = \begin{cases} 0 & m > 21000 \\ 0{,}1 + \frac{21000 - m}{90000} & 21000 \geq m > 3000 \\ 0{,}3 + \frac{21000 - 7m}{5000} & 3000 \geq m > 2500 \\ 1{,}0 & 2500 \geq m \end{cases}$$