

异或粽子 (xor)

【题目描述】

小粽是一个喜欢吃粽子的好孩子。今天她在家里自己做起了粽子。

小粽面前有 n 种互不相同的粽子馅儿，小粽将它们摆放为了一排，并从左至右编号为 1 到 n 。第 i 种馅儿具有一个非负整数的属性值 a_i 。每种馅儿的数量都足够多，即小粽不会因为缺少原料而做不出想要的粽子。小粽准备用这些馅儿来做出 k 个粽子。

小粽的做法是：选两个整数数 l, r ，满足 $1 \leq l \leq r \leq n$ ，将编号在 $[l, r]$ 范围内的所有馅儿混合做成一个粽子，所得的粽子的美味度为这些粽子的属性值的异或和。（异或就是我们常说的 xor 运算，即 C/C++ 中的 `^` 运算符或 Pascal 中的 `xor` 运算符）

小粽想品尝不同口味的粽子，因此它不希望用同样的馅儿的集合做出一个以上的粽子。

小粽希望她做出的所有粽子的美味度之和最大。请你帮她求出这个值吧！

【输入格式】

从文件 `xor.in` 中读入数据。

第一行两个正整数 n, k ，表示馅儿的数量，以及小粽打算做出的粽子的数量。

接下来一行为 n 个非负整数，第 i 个数为 a_i ，表示第 i 个粽子的属性值。

对于所有的输入数据都满足： $1 \leq n \leq 5 \times 10^5, 1 \leq k \leq \min\{\frac{n(n-1)}{2}, 2 \times 10^5\}, 0 \leq a_i \leq 4,294,967,295$ 。

【输出格式】

输出到文件 `xor.out` 中。

输出一行一个整数，表示小粽可以做出的粽子的美味度之和的最大值。

【样例 1 输入】

```
3 2
1 2 3
```

【样例 1 输出】

```
6
```

【样例 1 解释】

小粽可以选取 $[3, 3], [1, 2]$ 两个区间的馅儿做成粽子，两个粽子的美味度都为 3，和为 6。可以证明不存在比这更优的方案。

【样例 2】

见选手目录下的 *xor/xor2.in* 与 *xor/xor2.ans*。

【子任务】

测试点	n	k
1,2,3,4,5,6,7,8	$\leq 10^3$	$\leq 10^3$
9,10,11,12	$\leq 5 \times 10^5$	
13,14,15,16	$\leq 10^3$	$\leq 2 \times 10^5$
17,18,19,20	$\leq 5 \times 10^5$	