Round screws

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The NIT likes round screws. He also likes the $\oplus$ operator because it reminds him of round screws, where $\oplus$ represents Bitwise-XOR operation.

Define the value V_a of a sequence $a_1, a_2, \cdots, a_n$ as $V_a = a_1 + a_n + \sum_{i=1}^{n-1} (a_i \oplus a_{i+1})$.

Given a sequence $a_1, a_2, \cdots, a_n$, you can perform the following operation for arbitrary times:

- Select an index i ($1 \le i \le n$), change a_i to any non-negative integer; this operation has a cost C.

Minimize the sum of the value of the sequence and the cost incurred by operations. In other words, let p be the number of operations you performed, and $V_{a'}$ be the value of the a after the operations, then you need to minimize $pC + V_{a'}$.

Input

The input contains multiple testcases. The first line of the input contains an integer T ($1 \le T \le 100$), the number of testcases.

For each testcase, the first line contains two integers n, C ($2 \le n \le 10^5, 0 \le C < 2^{19}$), the length of the sequence and the cost of performing an operation.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($0 \le a_i < 2^{18}$), representing the elements in the sequence.

It's guaranteed that the sum of n over all testcases does not exceed 2×10^5.

Output

For each testcase, print an integer in one line, the minimum possible value of $pC + V_{a'}$.

Example

standard input	standard output
3 4 4 1 4 5 6 8 6 6 6 6 1 1 6 6 6 6 7 1 7 2 6 3 5	14 24 29

Note

For the first testcase, one way to achieve minimum is to change the sequence to $1, \mathbf{1}, \mathbf{1}, \mathbf{0}$; the bold numbers are the changed. The final sequence value is 2, and 3 operations are done; the total value + cost is $2 + 3 \times 4 = 14$.

For the second testcase, one way to achieve minimum is to change the sequence to $6, 6, 6, \mathbf{6}, \mathbf{6}, 6, 6, 6$; the final value is $12 + 2 \times 6 = 24$.