E. XOR on Segment

time limit per test: 4 seconds
memory limit per test: 256 megabytes

You've got an array a, consisting of n integers $a_1, a_2, ..., a_n$. You are allowed to perform two operations on this array:

1. Calculate the sum of current array elements on the segment $[l, r]$, that is, count value $a_l + a_{l+1} + ... + a_r$.
2. Apply the xor operation with a given number x to each array element on the segment $[l, r]$, that is, execute $a_l = a_l \oplus x, a_{l+1} = a_{l+1} \oplus x, \ldots, a_r = a_r \oplus x$. This operation changes exactly $r - l + 1$ array elements.

Expression $x \oplus y$ means applying bitwise xor operation to numbers x and y. The given operation exists in all modern programming languages, for example in language *C++* and *Java* it is marked as "^", in *Pascal* — as "xor".

You've got a list of m operations of the indicated type. Your task is to perform all given operations, for each sum query you should print the result you get.

Input

The first line contains integer n $(1 \le n \le 10^5)$ — the size of the array. The second line contains space-separated integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^6)$ — the original array.

The third line contains integer m $(1 \le m \le 5 \cdot 10^4)$ — the number of operations with the array. The i-th of the following m lines first contains an integer t_i $(1 \le t_i \le 2)$ — the type of the i-th query. If $t_i = 1$, then this is the query of the sum, if $t_i = 2$, then this is the query to change array elements. If the i-th operation is of type 1, then next follow two integers l_i, r_i $(1 \le l_i \le r_i \le n)$. If the i-th operation is of type 2, then next follow three integers l_i, r_i, x_i $(1 \le l_i \le r_i \le n, 1 \le x_i \le 10^6)$. The numbers on the lines are separated by single spaces.

Output

For each query of type 1 print in a single line the sum of numbers on the given segment. Print the answers to the queries in the order in which the queries go in the input.

Please, do not use the %lld specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams, or the %I64d specifier.

Examples

input Copy

```
5
4 10 3 13 7
8
1 2 4
2 1 3 3
1 2 4
1 3 3
2 2 5 5
1 1 5
2 1 2 10
1 2 3
```

output Copy

```
26
22
0
34
11
```

input Copy

```
6
4 7 4 0 7 3
5
2 2 3 8
1 1 5
2 3 5 1
2 4 5 6
1 2 3
```

output Copy

```
38
28
```

→ **Attention**

The package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, a solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, then the value 800 ms will be displayed and used to determine the verdict.

Codeforces Round 149 (Div. 2)

Finished

Practice

☆

→ **Virtual participation**

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ **Clone Contest to Mashup**

You can clone this contest to a mashup.

Clone Contest

→ **Submit?**

Language: GNU G++20 13.2 (64 bit, win

Choose file: Choose File No file chosen

Submit

→ **Last submissions**

Submission	Time	Verdict
291100038	Nov/11/2024 20:51	Accepted
291099984	Nov/11/2024 20:50	Wrong answer on test 2
291099884	Nov/11/2024 20:50	Wrong answer on test 1
291099689	Nov/11/2024 20:48	Wrong answer on test 1
271836333	Jul/22/2024 01:48	Accepted
271836256	Jul/22/2024 01:46	Wrong answer on test 1
271722704	Jul/21/2024 11:49	Accepted

→ **Problem tags**

bitmasks data structures *2000