Given an array of N numbers, we wish to choose a contiguous sub-sequence of the array, so that the bitwise XOR of all chosen numbers is maximum. Bitwise XOR is defined as follows: every bit in the answer is obtained by applying XOR logic on the corresponding bits of the set of numbers. For example 7, 8 and 5 are XORed as follows,

```
Numbers in binary:  0111  1000  0101  -----  1010
```

So the answer is 10 (in decimal). The same answer can be obtained in C/C++/Java by using the XOR operator as 7^8^5.

Input

The first line contains the number of test cases T. The first line of each test-case contains one integer, N (size of the array). The next N lines of each test-case contain integers denoting the elements of the array.

Output

For each test case, output a single line containing the maximum sum that can be obtained.

Constraints:

- $1 \leq T \leq 10$
- $1 \leq N \leq 100,000$
- All input integers will be non-negative and fit into 32 bit signed integer.

Sample Input

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

Sample Output

```
7  15
```