Hashing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In this problem you are given a byte array a. What you are going to do is to hash its subsequences. Fortunately you don't have to make a painful choice among infinitely large number of ways of hashing, as we have made this decision for you.

If we consider a subsequence as a strictly increasing sequence s of indices of array a, the hash function of the subsequence is calculated by the formula:

$$Hash(s) = \sum_{0 \le i < \text{sizeof}(s)} i \oplus a_{s_i}$$

Here, $\oplus$ means the bitwise XOR operation. See Note section if you need a clarification.

As you need to store the values in an array after all, you want to know the maximum possible value of the hash function among all subsequences of array a.

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$), denoting the number of bytes in array a. The second line contains n bytes written in hexadecimal numeral system and separated by spaces. Each byte is represented by exactly two hexadecimal digits (`0`...`F`).

Output

Output a single integer which is the maximum possible value of the hash function a subsequence of array a can have.

Examples

standard input	standard output
3 03 00 1B	29
3 01 00 02	4

Note

In the first sample one of the best ways is to choose the subsequence `03 00 1B`.

$hash = (0 \oplus 3) + (1 \oplus 0) + (2 \oplus 27) = 3 + 1 + 25 = 29$

In the second sample the only best way is to choose the subsequence `01 02`.

$hash = (0 \oplus 1) + (1 \oplus 2) = 1 + 3 = 4$

Here we are to tell you what a bitwise XOR operation is. If you have two integers x and y, consider their binary representations (possibly with leading zeroes): $x_k \dots x_2 x_1 x_0$ and $y_k \dots y_2 y_1 y_0$. Here, x_i is the i-th bit of number x and y_i is the i-th bit of number y. Let $r = x \oplus y$ be the result of XOR operation of x and y. Then r is defined as $r_k \dots r_2 r_1 r_0$ where:

$$r_i = \begin{cases} 1, & \text{if } x_i \ne y_i \\ 0, & \text{if } x_i = y_i \end{cases}$$