The Elias gamma code is a simple code which can be used to encode a sequence of positive integers. We will use a modified code which is also able to encode zeros. To encode an integer n, do the following:

1. Let k be the number of bits of n
2. Write $k-1$ zeros followed by a 1
3. Write n in binary

Examples

Number	Binary	Number of bits	Prefix	Code
0	0	1	1	10
1	1	1	1	11
2	10	2	01	0110
3	11	2	01	0111
4	100	3	001	001100
5	101	3	001	001101
6	110	3	001	001110
7	111	3	001	001111
8	1000	4	0001	00011000

A sequence of integers is encoded by writing the codes of the individual integers of the sequence in the same order as the integers appear in the sequence. The prefix of k additional bits before the binary representation of each integer is needed to be able to decode the encoded integers. So when reading the encoding of a sequence of integers, if we read $k-1$ zeros followed by a one, it means that there are k bits following which are the binary representation of the next encoded integer.

If we want to shorten the length of the encoding of a sequence of integers, there may be still some room for improvement; we will consider the following two optimizations:

1. If there is a prefix which indicates that k bits are following, but there is no integer in the sequence with k bits, we can use this prefix to indicate that $k+1$ bits are following. If there already was a prefix which indicates that $k+1$ bits are following, this prefix is not needed anymore, and it can be used to indicate that $k+2$ bits are following, and so on.
2. We can add a leading zero to the binary representation of all integers in the sequence with k bits, which then become integers with $k+1$ bits, and then the first optimization can be used. This optimization seems especially useful if there are few integers with k bits, but many integers with more than k bits.

When we are minimizing the length of the encoding of a sequence of integers, we only care about how many integers in the sequence have a certain number of bits. Let c_i denote the number of integers in a sequence with i bits.

Let us look at the following example: $c_1 = 2$, $c_2 = 4$, $c_3 = 0$, $c_4 = 1$ (which, for example, could correspond to a sequence 2, 1, 3, 8, 0, 2, 3). With the original elias gamma coding, the encoding of the sequence would have length $2 \times (1+1) + 4 \times (2+2) + 0 \times (3+3) + 1 \times (4+4) = 28$. By using optimization 1 we can save 1 bit by using prefix '`001`' for the integer with 4 bits. Then, we could use optimization 2 and add leading zeros to the integers with 1 bit, making them use 2 bits. Then, we use optimization 1 and use prefix 1 for the integers with 2 bits, prefix '`01`' for the integer with 4 bits, and we get the new length of $6 \times (1+2) + 1 \times (2+4) = 24$.

Both optimizations can possibly be used several times. Note that for the second optimization, it is not easy to decide when and how to use it. The goal is to combine these two optimizations in the best possible way, that means we want to find an encoding of a given sequence of integers that has minimum length among all encodings using elias gamma coding with any combination of these two optimizations.

Input

The input file contains several test cases. Each test case starts with a line containing an integer n, $(1 \le n \le 128)$. The next line contains the values $c_1, \ldots, c_n$ $(0 \le c_i \le 10000)$. Input is terminated by $n = 0$.

Output

For each test case print one line with the minimum length of an encoding of the given input sequence.

Note: The first sample test case corresponds to the example given in the problem description.

Sample Input

```
4
2 4 0 1
5
9 4 2 4 3
11
44 56 96 26 73 80 77 50 33 16 78
0
```

Sample Output

```
24
99
5494
```