7625 Rice coding

Variable length prefix-free codes are a great way to reduce the storage space required to store integers that have an unequal frequency distribution across a large range, i.e., if we expect small numbers to appear more frequently than larger numbers. Rice coding is a particular variable-length coding scheme that is both simple, and efficient to implement. A Rice code, parameterized by a *Rice parameter* k, comprises a *unary* part, and a *binary* part.

The *unary representation* of the non-negative integer i is just a sequence of i zero (0) bits followed by a one (1) bit, e.g., 5 is encoded as 000001_2 (the subscript "2" denotes the value is represented in base 2). The length of the unary encoding of i is $i + 1$ bits. Note that a unary code can be uniquely decoded by simply counting the number of leading 0 bits; a binary-coded integer cannot be parsed without knowing its length in advance.

To obtain the Rice coding of an integer i, we compute the quotient

$$q(i, k) = \left\lfloor \frac{i}{2k} \right\rfloor \tag{1}$$

where k is the Rice parameter. The unary part of the Rice code is then the value $q(i, k)$ encoded in a unary representation. Next, we compute the remainder

$$r = i - 2^k q(i, k)$$

which we can then encode as a binary representation in exactly k bits, which forms the binary part of the Rice code.

Let $L_k(i)$ denote the length (in bits) of the Rice encoding of the non-negative integer i using a Rice parameter of k, then

$$L_k(i) = q(i, k) + 1 + k.$$

Negative integers can be accommodated by the mapping function U,

$$U(i) = \begin{cases} -2i - 1 & \text{if} \quad i < 0 \\ 2i & \text{if} \quad i \geq 0, \end{cases}$$

so that the Rice coding length of a signed integer i is denoted $L_k(U(i))$.

If we want to encode 16-bit integers using Rice coding with a parameter $k = 3$, we find that $L_3(32765) = 4099$ bits, as opposed to the 16 bits we started with. To prevent unexpectedly large values (compared to our typical values) from producing such impractically long codes, we limit the value of $q(i, k)$ to an upper limit of 8. Whenever $q(i, k) \geq 8$, we output the value 8 in its unary representation, followed by the 16-bit binary representation of i, resulting in a length of $8 + 1 + 16$ bits. The length of this modified Rice code is thus:

$$M_k(i) = \begin{cases} L_k(U(i)) & \text{if} \quad q(U(i), k) < 8 \\ 8 + 1 + 16 & \text{if} \quad q(U(i), k) \geq 8, \end{cases} \tag{2}$$

Given a sequence of signed 16-bit integers, your task is to calculate the minimum possible length (in bits) of this sequence encoded using Rice coding where the Rice parameter k may be selected from the range [0, 14]. Note that a single value of k must be selected for the entire sequence.

Input

Your input consists of an arbitrary number of records, but no more than 100. Each record starts with a line containing a single integer n, with $1 \leq n \leq 5000$, denoting the number of integers to follow. The next line contains n integers s_i separated by spaces, subject to $-32768 \leq s_i \leq 32767$.

The end of input is indicated by a line containing only the value '`-1`'.

Output

For each input record, output the minimum number of bits required to encode the sequence of integers using Rice coding with a Rice parameter k selected from the range $[0, 14]$, with individual Rice code lengths as defined in Equation (2).

Sample Input

```
13
0 -1 -13807 -1 0 0 0 -2 0 -1 0 -1 0
16
1 -9 -7 -25 19 -2 -24 3 -60 -19 -27 -8 80 188 -17 -25
-1
```

Sample Output

```
44
120
```