Decoding of Varints

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Varint is a type used to serializing integers using one or more bytes. The key idea is to have smaller values being encoded with a smaller number of bytes.

First, we would like to encode some unsigned integer x. Consider its binary representation $x = a_0a_1a_2 \ldots a_{k-1}$, where a_i-th stands for the i-th significant bit, i.e. $x = a_0 \cdot 2^0 + a_1 \cdot 2^1 + \ldots + a_{k-1} \cdot 2^{k-1}$, while $k-1$ stands for the index of the most significant bit set to 1 or $k = 1$ if $x = 0$.

To encode x we will use $m = \lceil \frac{k}{7} \rceil$ bytes $b_0, b_1, \ldots, b_{m-1}$. That means one byte for integers from 0 to 127, two bytes for integers from 128 to $2^{14} - 1 = 16383$ and so on, up to ten bytes for $2^{64} - 1$. For bytes $b_0, b_1, \ldots, b_{m-2}$ the most significant bit is set to 1, while for byte b_{m-1} it is set to 0. Then, for each i from 0 to $k-1$, $i \ mod \ 7$ bit of byte $b_{\lfloor \frac{i}{7} \rfloor}$ is set to a_i. Thus,

$$x = (b_0 - 128) \cdot 2^0 + (b_1 - 128) \cdot 2^7 + (b_2 - 128) \cdot 2^{14} + \ldots + (b_{m-2} - 128) \cdot 2^{7 \cdot (m-2)} + b_{m-1} \cdot 2^{7 \cdot (m-1)}$$

In the formula above we subtract 128 from $b_0, b_1, \ldots, b_{m-2}$ because their most significant bit was set to 1.

For example, integer 7 will be represented as a single byte $b_0 = 7$, while integer 260 is represented as two bytes $b_0 = 132$ and $b_1 = 2$.

To represent signed integers we introduce *ZigZag* encoding. As we want integers of small magnitude to have short representation we map signed integers to unsigned integers as follows. Integer 0 is mapped to 0, -1 to 1, 1 to 2, -2 to 3, 2 to 4, -3 to 5, 3 to 6 and so on, hence the name of the encoding. Formally, if $x \geq 0$, it is mapped to $2x$, while if $x < 0$, it is mapped to $-2x - 1$.

For example, integer 75 is mapped to 150 and is encoded as $b_0 = 150$, $b_1 = 1$, while -75 will be mapped to 149 and will be encoded as $b_0 = 149$, $b_1 = 1$. In this problem we only consider such encoding for integers from -2^{63} to $2^{63} - 1$ inclusive.

You are given a sequence of bytes that corresponds to a sequence of signed integers encoded as varints. Your goal is to decode and print the original sequence.

Input

The first line of the input contains one integer n ($1 \leq n \leq 10\,000$) — the length of the encoded sequence. The next line contains n integers from 0 to 255. You may assume that the input is correct, i.e. there exists a sequence of integers from -2^{63} to $2^{63} - 1$ that is encoded as a sequence of bytes given in the input.

Output

Print the decoded sequence of integers.

Example

standard input	standard output
5 0 194 31 195 31	0 2017 -2018