

E. Bear and Bad Powers of 42

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak, a bear, isn't good at handling queries. So, he asks you to do it.

We say that powers of 42 (numbers 1, 42, 1764, ...) are *bad*. Other numbers are *good*.

You are given a sequence of n good integers t_1, t_2, \dots, t_n . Your task is to handle q queries of three types:

- 1 i — print t_i in a separate line.
- 2 a b x — for set t_i to x . It's guaranteed that x is a good number.
- 3 a b x — for increase t_i by x . After this repeat the process while at least one t_i is bad.

You can note that after each query all t_i are good.

Input

The first line of the input contains two integers n and q ($1 \leq n, q \leq 100\,000$) — the size of Limak's sequence and the number of queries, respectively.

The second line of the input contains n integers t_1, t_2, \dots, t_n ($2 \leq t_i \leq 10^9$) — initial elements of Limak's sequence. All t_i are good.

Then, q lines follow. The i -th of them describes the i -th query. The first number in the line is an integer $type_i$ ($1 \leq type_i \leq 3$) — the type of the query. There is at least one query of the first type, so the output won't be empty.

In queries of the second and the third type there is $1 \leq a \leq b \leq n$.

In queries of the second type an integer x ($2 \leq x \leq 10^9$) is guaranteed to be good.

In queries of the third type an integer x ($1 \leq x \leq 10^9$) may be bad.

Output

For each query of the first type, print the answer in a separate line.

Example

input
6 12 40 1700 7 1672 4 1722 3 2 4 42 1 2 1 3 3 2 6 50 1 2 1 4 1 6 2 3 4 41 3 1 5 1 1 1 1 3 1 5
output
1742 49 1842 1814

1822
43
44
107

Note

After a query 3 2 4 42 the sequence is 40, 1742, 49, 1714, 4, 1722.

After a query 3 2 6 50 the sequence is 40, 1842, 149, 1814, 104, 1822.

After a query 2 3 4 41 the sequence is 40, 1842, 41, 41, 104, 1822.

After a query 3 1 5 1 the sequence is 43, 1845, 44, 44, 107, 1822.