

Beats

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Nirva é um dos mais novos talentos da Unballoon. Como todos sabem, um dos principais conselhos que ouvimos no início da carreira em programação competitiva é focar em melhorar o raciocínio e deixar para aprender conteúdos mais avançados, como autômato de sufixos, fluxo, FFT etc., para depois. Como todos também sabem, quase sempre esse conselho é ignorado completamente. Por isso, Nirva decidiu aprender a “Segment Tree Beats”. O próprio autor dessa estrutura de dados tem isso a dizer sobre ela:

“In China, all of the 15 candidates for the Chinese National Team are asked to write a simple research report about algorithms in informatics Olympiad, and the score will be counted in the final selection. There are many interesting ideas and algorithms in these reports. And I find that some of them are quite new for competitors in CF although they are well known in China [...].

This blog is about my report which is written about two years ago. [...] the name “Segment tree beats” is given by C_SUNSHINE which is from a famous Japanese anime Angel Beats.”

Como esperado, Nirva entendeu facilmente o funcionamento da Segment Tree Beats e fez sua própria implementação. Para testar se ela está correta, ela achou que seria interessante comparar a implementação dela com a sua.

Você recebe n inteiros não negativos a_1, \dots, a_n . Seu programa deve processar q operações de três tipos:

1. $\text{chmod}(l, r, x)$ — Para todos os índices i tais que $l \leq i \leq r$, substitua o valor de a_i por $a_i \% x$.
2. $\text{chmax}(l, r, x)$ — Para todos os índices i tais que $l \leq i \leq r$, substitua o valor de a_i por $\max(a_i, x)$.
3. $\text{soma}(l, r)$ — Imprima a soma dos elementos a_i tais que $l \leq i \leq r$, isto é, $a_l + a_{l+1} + \dots + a_r$.

Input

A primeira linha contém dois inteiros n e q ($1 \leq n, q \leq 1000$). A segunda linha contém n inteiros a_1, \dots, a_n ($0 \leq a_i \leq 10^6$).

Cada uma das próximas q linhas descreve uma operação em um dos formatos:

- $1 \ l \ r \ x$ — representando a operação $\text{chmod}(l, r, x)$;
- $2 \ l \ r \ x$ — representando a operação $\text{chmax}(l, r, x)$;
- $3 \ l \ r$ — representando a operação $\text{soma}(l, r)$.

É garantido que, em cada operação, $1 \leq l \leq r \leq n$ e $1 \leq x \leq 10^6$.

Output

Para cada operação do tipo 3 (soma), imprima um único inteiro representando o resultado dessa operação.

Example

standard input	standard output
6 5	15
1 2 3 4 5 6	9
3 1 5	4
1 2 4 3	
3 1 5	
2 1 4 2	
3 2 3	