E1. Summer Homework

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

By the age of three Smart Beaver mastered all arithmetic operations and got this summer homework from the amazed teacher:

You are given a sequence of integers $a_1, a_2, ..., a_n$. Your task is to perform on it m consecutive operations of the following type:

1. For given numbers x_i and v_i assign value v_i to element a_{x_i}.
2. For given numbers l_i and r_i you've got to calculate sum , where $f_0 = f_1 = 1$ and at $i \geq 2$: $f_i = f_{i-1} + f_{i-2}$.
3. For a group of three numbers $l_i\ r_i\ d_i$ you should increase value a_x by d_i for all x $(l_i \leq x \leq r_i)$.

Smart Beaver planned a tour around great Canadian lakes, so he asked you to help him solve the given problem.

Input

The first line contains two integers n and m $(1 \leq n, m \leq 2 \cdot 10^5)$ — the number of integers in the sequence and the number of operations, correspondingly. The second line contains n integers $a_1, a_2, ..., a_n$ $(0 \leq a_i \leq 10^5)$. Then follow m lines, each describes an operation. Each line starts with an integer t_i $(1 \leq t_i \leq 3)$ — the operation type:

- if $t_i = 1$, then next follow two integers $x_i\ v_i$ $(1 \leq x_i \leq n, 0 \leq v_i \leq 10^5)$;
- if $t_i = 2$, then next follow two integers $l_i\ r_i$ $(1 \leq l_i \leq r_i \leq n)$;
- if $t_i = 3$, then next follow three integers $l_i\ r_i\ d_i$ $(1 \leq l_i \leq r_i \leq n, 0 \leq d_i \leq 10^5)$.

The input limits for scoring 30 points are (subproblem E1):

- It is guaranteed that n does not exceed 100, m does not exceed 10000 and there will be no queries of the 3-rd type.

The input limits for scoring 70 points are (subproblems E1+E2):

- It is guaranteed that there will be queries of the 1-st and 2-nd type only.

The input limits for scoring 100 points are (subproblems E1+E2+E3):

- No extra limitations.

Output

For each query print the calculated sum modulo 1000000000 (10^9).

Examples

input
5 5 1 3 1 2 4 2 1 4 2 1 5 2 2 4 1 3 10 2 1 5

output
12 32

```
8
50
```

input

```
5 4
1 3 1 2 4
3 1 4 1
2 2 4
1 2 10
2 1 5
```

output

```
12
45
```