We define a *modified Fibonacci sequence* using the following definition:

Given terms t_i and t_{i+1} where $i \in [1, \infty)$, term t_{i+2} is computed using the following relation:

$$t_{i+2} = t_i + (t_{i+1})^2$$

For example, if term $t_1 = 0$ and $t_2 = 1$, term $t_3 = 0 + 1^2 = 1$, term $t_4 = 1 + 1^2 = 2$, term

$t_5 = 1 + 2^2 = 5$, and so on.

Given three integers t_1, t_2, and n, compute and print term t_n of our *modified Fibonacci sequence*.

Note: The value of t_n may exceed the range of a 64-bit integer. Since the value will always be non-negative, it will be wise to use the "`unsigned long long`" datatype. Another smart strategy would be to mod each term of the formula by the macro "`ULLONG_MAX`".

Input

A single line of three space-separated integers describing the respective values of t_1, t_2, and n. Here:

- $0 \leq t_1, t_2 \leq 2$
- $3 \leq n \leq 20$
- t_n may exceed the range of a 64-bit integer

Output

Print a single integer denoting the value of term t_n in the modified Fibonacci sequence where the first two terms are t_1 and t_2.

Sample Input	Sample Output
0 1 5	5