

Given an array containing  $n$  integers. You need to find the  $k$ -th order statistics in this array, that is, the element that, after sorting the array in non-descending order, will be at the  $k$ -th place from the beginning of the array (element indexing starts from zero).

The only line of the input contains three integers  $n$ ,  $a_0$  and  $k$  are the number of elements in the array, the value of the element with index 0, and the number of the required order statistics ( $1 \leq n \leq 2 \cdot 10^7$ ;  $0 \leq a_0 < 2^{31}$ ;  $0 \leq k < n$ ).

The remaining elements of the array must be generated. Array elements are set using a pseudo-random generator according to the formula:  $a_i = (1103515245 \cdot a_{i-1} + 12345) \bmod 2^{31}$ . To fill array elements with initial values, you can use the following C++ function:

```
void fill(vector<int>& a, int n, int start)
{
    a.resize(n);
    a[0] = start;
    for (int i = 1; i < n; ++i) {
        a[i] = (1103515245LL * a[i - 1] + 12345) % (1LL << 31);
    }
}
```

The program should output one integer – the  $k$ -th minimum in the given sequence.

---

**Sample input:**

5 123456789 2

**Sample output:**

850994577