Optimal cache

Implement an optimal cache replacement algorithm.

Input

The input consists of two lines. The first of these lines gives the size of the cache. The second of these lines describes a sequence of memory accesses. The sequence is described by N followed by x1, ..., xN: N is the length of the sequence, and each xk is a memory address.

The input is made out of whitespace-separated integers. All integers are positive and at most a million.

Output

The number of cache faults.

Example

Input:

```
2
4 1 2 3 1
```

Output:

```
3
```

We are supposed to simulate an *optimal* cache of size 2. The first two memory accesses cause faults, and locations 1 and 2 are loaded into the cache. The third access, to address 3, causes a fault as well, and location 3 is loaded into the cache; but, first, to make room for the value at address 3, and an optimal cache would evict location 2. The fourth access does not cause a fault because location 1 is still in the cache.

In total, there are 3 faults.

See also: FIFOCACH and LRUCACHE.