

FIFO cache

Implement the FIFO 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 x_1, \dots, x_N : N is the length of the sequence, and each x_k 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 2
```

Output:

```
3
```

We are supposed to simulate a FIFO 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, location 1 is evicted. Location 1 is evicted because it is the oldest one to have been loaded into the cache -- this is what FIFO means. Finally, the fourth access does not cause a fault, because location 2 is still in the cache.

In total, there are 3 faults.

See also: LRUCACHE and MINCACHE.