4016   David Shopping

David comes to Chengdu for ACM-ICPC 2007. After learning Chengdu is a beautiful city, David decides to buy his friends some gifts.

The capacity of David's pocket is so small that it can only contain M gifts. Considering the diversity of his gifts, David would not buy two of the same kind. And some typical gift should be chosen to represent the features of Chengdu.

David will walk down from north to south and visit N shops one by one along the shopping street. There is *ONLY ONE* type of gift sold in each shop.

David has such a poor memory that he can't remember how many shops sell gift K. So he will write a number L on this gift he bought in his pocket, to indicate how many shops where sell gift K. In David's opinion, the smaller the number L is, the better the gift (David like uncommon gifts).

When David stops in a shop which sells gift K, the following three situations he might come across.

1. If there is not gift K in his pocket and still some place for it, He will buy without hesitation. Before putting it into the pocket, David will write down the number '1' on the gift to indicate that he has once seen one shop selling it.

2. If there is gift K in his pocket, David will just replace the number L with $L+1$, indicating $L+1$ shops sell gift K.

3. If there is not gift K in his pocket and the pocket is full, David would like to regard no shops selling gift K (because he cannot remember whether or not he has met gift K), so he will have to discard one gift in his pocket to release a place for the gift K. But which gift should be discarded? According to the follow rule:

   He chooses the gift that has the biggest number L on it. If several gifts have the same biggest number L, he will discard the one which has been putted into the pocket at the earliest time. After discarding the gift, he will put gift K into his pocket and write number '1' on gift K.

Now, your task is to write a program to record the number of these gifts which have been discarded by David.

For example:

David's pocket has the capacity only for two gifts. There are 5 shops in the street, and each shop sells only one type of gift. The selling sequence of gifts is 1, 2, 1, 3, and 1.

In shop 1, the pocket is empty, so he will buy gift 1, write a number '1' on this gift, and then put it into his pocket.

When he comes to shop 2, there is one place left in his pocket, so he buys gift 2, write a number '1' on it, and then put it into the pocket.

When walking into shop 3, he has already got gift 1 in his pocket, so he will replace the number L (here, $L=1$) with $L+1$.

When David visits shop 4, the pocket is full, but without gift 3 in it, so he has to discard one gift to release a place for gift 3. The number L on gift 1 is '2', but the number L on gift 2 is '1', so he will discard gift 1, write number 1 on gift 3 and then put it into the pocket.

In shop 5, the pocket is full, gift 1 is not in it, he should will discard a gift to find place for gift 1. The number L on gift 2 is '1', the number L on gift 3 is also '1'. They have the same L, but gift 2 put into the pocket earlier than gift 3. So he discards gift 2, write number '1' on gift 1 and then put it into the pocket.

At the end of the street, David gets two gifts in his pocket, number '1' on gift 3 and number '1' on gift 1. The number of discarded gifts is 2.

Input

There are multiple test cases in the input file. Each test case contains two lines.

The first line has two positive integers M and N ($M \leq 50,000$ and $N \leq 100,000$) where M (the capacity of pocket) shows how many gifts it can take and N is the number of shops in the street. The second line has N positive integers K_i ($K_i < 2^{20}, i = 1, 2, \ldots, N$) indicating the type of gift sold in the i-th shop. $M = 0$ and $N = 0$ indicate the end of file and should not be processed by your program.

Output

For each test case you should output one integer, the number of discarded gifts as indicated in the sample output.

Sample Input

```
3 5
1 2 3 2 4
2 4
1 2 2 1
2 6
1 2 2 1 1024 1
2 10
1 2 3 2 4 2 3 6 7 8
2 1
1048575
6 16
10 1 2 3 4 5 6 1 2 3 6 5 4 10 1 6
0 0
```

Sample Output

```
Case 1: 1
Case 2: 0
Case 3: 2
Case 4: 7
Case 5: 0
Case 6: 3
```