

Problem C

Smart Thief

Ayu managed to steal Budi's treasure box and ready to uncover any secret Budi hides. Unfortunately, the treasure box has some security system to prevent any unauthorized person (e.g., Ayu) from opening it.

To unlock the treasure box, Ayu has to input a correct PIN (Personal Identification Number) of length N , which of course, she doesn't have. Ayu has no choice other than trying all possible PIN combinations. However, Ayu notices that this security system has an interesting (old) mechanism.

When you enter an N digits PIN, it is evaluated automatically and promptly, i.e. you don't need to push some "enter" button to confirm the PIN. Whenever your entered PIN is wrong, it removes the oldest (first) digit and shifts all the remaining to the left, thus, you only need to enter one more (last) digit to make it N length again.

For example, let $N = 4$. If we input 204320435, then we actually test 6 PINs (with 5 different PINs):

- [2043] 20435, tested PIN = 2043
- 2[0432] 0435, tested PIN = 0432
- 20[4320] 435, tested PIN = 4320
- 204[3204] 35, tested PIN = 3204
- 2043[2043] 5, tested PIN = 2043
- 20432[0435], tested PIN = 0435

Notice that 2043 is tested twice in this example.

As a CS student, Ayu knows that finding the correct PIN by trying all possible combinations can be very time-consuming, but alas, there's no other way. Ayu decides that she wants to test at least K different PINs on the first day. Your task is to help Ayu by simply giving her the string S which contains at least K different PINs. Ayu doesn't care which PIN she's going to test (so long at least there are K different PINs) or whether any PIN is tested more than once in S , but string S needs to be as short as possible. If there is more than one possible string for S , you can output any of them.

Note that, as this system is quite old, there are only M available digits ranging from 0 to 9.

Input

Input begins with a line containing three integers: $N M K$ ($1 \leq N \leq 100000$, $1 \leq M \leq 10$, $1 \leq K \leq \min(M^N, 100000)$), representing the PIN length, the number of available digits, and the minimum number of PINs to be tested, respectively. The next line contains M integers: A_i ($0 \leq A_i \leq 9$), representing the available digits. You may assume all A_i are distinct. You may also assume that the values of N , M , and K are chosen such that the answer contains no longer than 100000 digits

Output

Output in a line the **shortest** string which contains at least K different PINs as its substring. If there is more than one such string, you can output any of them.

Sample Input #1

```
3 2 5
4 7
```

Sample Output #1

```
7477447
```

Explanation for the sample input/output #1

There are 5 different PINs of length 3 tested with the string 7477447, i.e. 447, 477, 744, 747, and 774.

Sample Input #2

```
2 5 9
1 2 3 4 5
```

Sample Output #2

```
1234554321
```

Explanation for the sample input/output #2

There are 9 different PINs of length 2 tested with the string 1234554321, i.e. 12, 21, 23, 32, 34, 43, 45, 54, and 55.

Sample Input #3

```
6 3 2
9 3 5
```

Sample Output #3

```
9353593
```

Explanation for the sample input/output #3

There are 2 different PINs of length 6 tested with the string 9353593, i.e. 353593, and 935359.