

# CCC '14 S1 - Party Invitation

---

**Time Limit:** 2.0s    **Memory Limit:** 64M

---

## Canadian Computing Competition: 2014 Stage 1, Junior #4, Senior #1

You are hosting a party and do not have room to invite all of your friends. You use the following unemotional mathematical method to determine which friends to invite.

Number your friends  $1, 2, \dots, K$  and place them in a list in this order. Then perform  $m$  rounds. In each round, use a number to determine which friends to remove from the ordered list.

The rounds will use numbers  $r_1, r_2, \dots, r_m$ . In round  $i$  remove all the remaining people in positions that are multiples of  $r_i$  (that is,  $r_i, 2r_i, 3r_i, \dots$ ). The beginning of the list is position 1.

Output the numbers of the friends that remain after this removal process.

## Input Specification

---

The first line of input contains the integer  $K$  ( $1 \leq K \leq 100$ ). The second line of input contains the integer  $m$  ( $1 \leq m \leq 10$ ), which is the number of rounds of removal. The next  $m$  lines each contain one integer. The  $i^{th}$  of these lines ( $1 \leq i \leq m$ ) contains  $r_i$  ( $2 \leq r_i \leq 100$ ) indicating that every person at a position which is multiple of  $r_i$  should be removed.

## Output Specification

---

The output is the integers assigned to friends who were not removed. One integer is printed per line in increasing sorted order.

## Sample Input

---

```
10
2
2
3
```

## Output for Sample Input

---

```
1
3
7
9
```

## Explanation of Output for Sample Input

---

Initially, our list of invitees is `{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`. There will be two rounds of removals. After the first round of removals, we remove the even positions (i.e., every second position), which causes our list of invitees to be `{1, 3, 5, 7, 9}`. After the second round of removals, we remove every 3rd remaining invitee: thus, we keep 1 and 3, remove 5 and keep 7 and 9, which leaves us with an invitee list of `{1, 3, 7, 9}`.