

数组拼接题目描述：

现在有多组整数数组，需要将它们合并成一个新的数组。合并规则，从每个数组里按顺序取出固定长度的内容合并到新的数组中，取完的内容会删除掉，如果该行不足固定长度或者已经为空，则直接取出剩余部分的内容放到新的数组中，继续下一行。

输入描述：

第一行是每次读取的固定长度， $0 < \text{长度} < 10$

第二行是整数数组的数目， $0 < \text{数目} < 1000$

第 $3-n$ 行是需要合并的数组，不同的数组用回车换行分隔，数组内部用逗号分隔，最大不超过 100 个元素。

输出描述：

输出一个新的数组，用逗号分隔。

补充说明：

示例 1

输入：

```
3
2
2, 5, 6, 7, 9, 5, 7
1, 7, 4, 3, 4
```

输出：

```
2, 5, 6, 1, 7, 4, 7, 9, 5, 3, 4, 7
```

说明：

1、获得长度 3 和数组数目 2 。

2、先遍历第一行，获得 $2, 5, 6$ ；

3、再遍历第二行，获得 1,7,4;

4、再循环回到第一行，获得 7,9,5;

5、再遍历第二行，获得 3,4;

6、再回到第一行，获得 7，按顺序拼接成最终结果。

示例 2

输入:

```
4
3
1,2,3,4,5,6
1,2,3
1,2,3,4
```

输出:

```
1,2,3,4,1,2,3,1,2,3,4,5,6
```

说明:

```
import java.util.*;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int m = Integer.parseInt(sc.nextLine());
        int n = Integer.parseInt(sc.nextLine());
        List<List<Integer>> arrs = new ArrayList<>();
        int count = 0;
        for (int i = 0; i < n; i++) {
            String tmp[] = sc.nextLine().split(",",-1);
            List<Integer> item = new ArrayList<>();
            for (String s : tmp) {
                item.add(Integer.parseInt(s));
            }
            arrs.add(item);
            count += item.size();
        }

        List<Integer> res = new ArrayList<>();
```

```

while (count != res.size()) {
    for (int i = 0; i < arrs.size(); i++) {
        int tmp = 0;
        for (int j = 0; j < arrs.get(i).size(); j++) {
            if (arrs.get(i).get(j) != -1) {
                res.add(arrs.get(i).get(j));
                arrs.get(i).set(j, -1);
                tmp += 1;
            }
            if (tmp == m) {
                break;
            }
        }
    }
}

List<String> ret = new ArrayList<>();
for (Integer re : res) {
    ret.add(re + "");
}

System.out.println(String.join(", ", ret));
}

}

```