

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

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：

```
#include<bits/stdc++.h>
```

```
#include <iostream>
```

```
using namespace std;
```

```
vector<int> v[100010];
```

```
int main() {
```

```
    int n;
```

```
    cin >> n;
```

```
    string s;
```

```
    int idx = 0;
```

```
    int sum = 0;
```

```
    int k;
```

```
    cin >> k;
```

```
    while (cin >> s) {
```

```
        ++idx;
```

```
        s += ',';
```

```
        int num = 0;
```

```
        for (auto j : s) {
```

```
            if (j == ',')
```

```
                v[idx].push_back(num), num = 0;
```

```
            else
```

```
                num = num * 10 + j - '0';
```

```
        }
```

```
        reverse(v[idx].begin(), v[idx].end());
```

```
        sum += v[idx].size();
```

```
    }
```

```

int pos = 0;
vector<int>ans;
while (sum > 0) {
    for (int j = 0; j < n; j++) {
        if (v[pos].size()) {
            ans.push_back(v[pos].back());
            v[pos].pop_back();
            sum--;
        }
    }
    pos += 1;
    if (pos == idx + 1)
        pos = 1;

}
for (int i = 0; i < ans.size(); i++) {
    if (i)
        cout << ",";
    cout << ans[i];
}
}

```