

### C++-数组字符串-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， $0 < \text{长度} < \text{数目}$

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。

#### 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：

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 vector<int> strToVec(const string& s) {
6     string sub;
7     vector<int> v;
8     for (int i = 0; i < s.size(); i++) {
9         const char& c = s[i];
10        if (isdigit(c)) {
11            sub.append(1, c);
12        }
13        else {
14            v.push_back(stoi(sub));
15            sub.clear();
16        }
17    }
18    v.push_back(stoi(sub));
19    sub.clear();
20
21    return std::move(v);
22 }
23
24
```

```

25 int main() {
26     int readLen;
27     int arrNum;
28     vector<vector<int>> arrs;
29     vector<int> indexes;
30     vector<int> finalArr;
31
32     cin >> readLen;
33     cin >> arrNum;
34
35     while (arrNum--> {
36         string s;
37         cin >> s;
38         if(!s.empty()){
39             arrs.push_back(strToVec(s));
40             indexes.push_back(0);
41         }
42     }
43
44     bool flag = true;
45     while (flag) {
46
47         flag = false;
48
49         for (int i = 0; i < arrs.size(); i++) {
50
51             vector<int> &v = arrs[i];
52
53             if (indexes[i] >= v.size()) {
54                 continue;
55             }
56
57             flag = true;
58         }
59     }

```

```

59     int j = 0;
60     for (; indexes[i] + j < v.size() && j < readLen; j++) {
61
62         int index = indexes[i] + j;
63
64         finalArr.push_back(v[index]);
65
66     }
67     indexes[i] += j;
68
69 }
70
71 }
72
73 for (int i = 0; i < finalArr.size(); i++) {
74     cout << finalArr[i];
75     if (i != finalArr.size() - 1) {
76         cout << ' ';
77     }
78 }
79 cout << endl;
80 }
81 // 64 位输出请用 printf("%lld")

```