

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。

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
```

```
#include <stdio.h>
```

```
long long A[1002][102];
```

```
char S[2002];
```

```
int indices[1002];
```

```
int C[1002];
```

```
long long R[100002];
```

```
int main()
```

```
{
```

```
    int i, j;
```

```
    int t;
```

```
    int L;
```

```
    int n;
```

```
    int N;
```

```
    int sign;
```

```
    long long v;
```

```
    int count = 0;
```

```
    scanf("%d", &L);
```

```
    scanf("%d", &n);
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        C[i] = 0;
```

```
    }
```

```

for (i = 0; i < n; i++)
{
    scanf("%s", S);
    sign = 1;
    v = 0;
    for (j = 0; S[j]; j++)
    {
        if (S[j] == ',')
        {
            v /= 10;
            A[i][C[i]] = sign * v;
            sign = 1;
            v = 0;
            C[i]++;
        }
        else
        {
            if (S[j] == '-')
            {
                sign = -1;
            }
            else
            {
                v += S[j] - '0';
                v *= 10;
            }
        }
    }
    v /= 10;
    A[i][C[i]] = sign * v;
    C[i]++;
}
N = 0;
for (i = 0; i < n; i++)
{
    N += C[i];
}
for (i = 0; i < n; i++)
{
    indices[i] = 0;
}
t = 0;
while (count < N)
{

```

```

    for (i = 0; i < L; i++)
    {
        if (indices[t] == C[t])
        {
            break;
        }
        R[count] = A[t][indices[t]];
        indices[t]++;
        count++;
        if (count == N)
        {
            break;
        }
    }
    t++;
    t %= n;
}
if (N)
{
    printf("%lld", R[0]);
}
for (i = 1; i < N; i++)
{
    printf(",%lld", R[i]);
}
return 0;
}

```