

```

1  /*
2  MATLAB中，有一个非常有用的函数 reshape，它可以将一个矩阵重塑为另一个大小不同的新矩阵，
   但保留其原始数据。
3
4  给出一个由二维数组表示的矩阵，以及两个正整数r和c，分别表示想要的重构的矩阵的行数和列数。
5
6  重构后的矩阵需要将原始矩阵的所有元素以相同的行遍历顺序填充。
7
8  如果具有给定参数的reshape操作是可行且合理的，则输出新的重塑矩阵；否则，输出原始矩阵。
9
10 示例 1:
11
12 输入:
13  nums =
14  [[1,2],
15   [3,4]]
16  r = 1, c = 4
17 输出:
18  [[1,2,3,4]]
19 解释:
20 行遍历nums的结果是 [1,2,3,4]。新的矩阵是 1 * 4 矩阵，用之前的元素值一行一行填充新矩阵。
21 示例 2:
22 输入:
23  nums =
24  [[1,2],
25   [3,4]]
26  r = 2, c = 4
27 输出:
28  [[1,2],
29   [3,4]]
30 解释:
31 没有办法将 2 * 2 矩阵转化为 2 * 4 矩阵。 所以输出原矩阵。
32 注意:
33
34     给定矩阵的宽和高范围在 [1, 100]。
35     给定的 r 和 c 都是正数。
36 来源：力扣（LeetCode）
37 链接：https://leetcode-cn.com/problems/reshape-the-matrix
38 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
39 */

```

分析:

- 按行遍历顺序进行取出和放入即可.

方法一:C++\_行遍历取出放入

```

1  class Solution
2  {

```

```

3      public:
4      vector<vector<int>> matrixReshape(vector<vector<int>>& nums, int r,
int c)
5      {
6          int rr = nums.size() ;
7          int cc = nums[0].size() ;
8          int i = 0 ;
9          vector<vector<int>> vvi ;
10
11
12         if(rr*cc == r*c)
13         {
14             return nums;
15         }
16         else
17         {
18             for( i = 0 ; i < r ; i++)
19             {
20                 vector<int> vi(c,0);
21                 vvi.push_back(vi);
22             }
23
24             for( i = 0 ; i < rr*cc ; i++)
25             {
26                 vvi[i/c][i%c] = nums[i/cc][i%cc];
27             }
28             return vvi;
29         }
30     }
31 };
32
33 /*
34 执行结果:
35 通过
36 显示详情
37 执行用时 :36 ms, 在所有 cpp 提交中击败了97.96% 的用户
38 内存消耗 :12.6 MB, 在所有 cpp 提交中击败了14.29%的用户
39 */

```