

**InfyTQ SET 003**Solved Challenges **4/5**[Back To Challenges List](#)**Submatrices - Sum of Digits****ID:10569   Solved By 212 Users****InfyTQ**

The program must accept an integer matrix of size **RxC** as the input. The program must print all possible **2x2** submatrices where each integer should follow the below rule.

- Each integer of the submatrix should be divisible by the sum of its digits.

**Note:** At least one such submatrix is always present in the given matrix.

**Boundary Condition(s):**

$2 \leq R, C \leq 50$

**Input Format:**

The first line contains R and C separated by a space.

The next R lines each contain C integers separated by a space.

**Output Format:**

The lines containing the 2x2 submatrices as per the given condition.

**Example Input/Output 1:**

Input:

```
4 4
18 19 72 42
92 84 60 63
12 50 93 35
24 54 94 37
```

Output:

```
72 42
60 63
12 50
24 54
```

Explanation:

The **2x2** submatrices where each integer of the submatrix is divisible by the sum of its digits are highlighted below.

```
18 19 72 42
92 84 60 63
12 50 93 35
24 54 94 37
```

**Example Input/Output 2:**

Input:

```
4 3
40 42 2
```

30 24 27  
180 190 40  
11 121 13

Output:  
40 42  
30 24  
30 24  
180 190  
24 27  
190 40

Max Execution Time Limit: 500 millisecs

Ambiance

Python3 (3.x )



Reset

```
1  R,C = map(int,input().strip().split())
2
3  matrix=[]
4  for row in range(R):
5      matrix.append(list(map(int,input().strip().split())))
6
7
8  def divisibility_check(sub_matrix):
9      count=0
10     for row in range(2):
11         for col in range(2):
12             num = sub_matrix[row][col]
13             n = num
14             add = 0
15             while(n>0):
16                 add+=(n%10)
17                 n//=10
18                 if(num%add==0):
19                     count+=1
20     if(count==4):
21         return 1
22     else:
23         return 0
24
25  sub_matrix = []
26  ans=[]
27  for row in range(R-1):
28      for col in range(C-1):
29          sub_matrix = []
30          for r in range(row,row+2):
31              li=[]
32              for c in range(col,col+2):
33                  li.append(matrix[r][c])
34              sub_matrix.append(li)
35          # print(sub_matrix)
36
37          if(divisibility_check(sub_matrix)==1):
38              ans.append(sub_matrix)
39
40
41  for sub in ans:
42      for r in range(2):
43          for c in range(2):
44              print(sub[r][c], end=" ")
45          print()
46
47
48
49
50
```

**Code did not pass the execution**

— ×

**TestCase ID: 58123****Input:**

```
4 4
18 19 72 42
92 84 60 63
12 50 93 35
24 54 94 37
```

**Expected Output:**

```
72 42
60 63
12 50
24 54
```

**Your Program Output:**

```
[[18, 19], [92, 84]]
[[19, 72], [84, 60]]
[[72, 42], [60, 63]]
[[92, 84], [12, 50]]
[[84, 60], [50, 93]]
[[60, 63], [93, 35]]
[[12, 50], [24, 54]]
[[50, 93], [54, 94]]
[[93, 35], [94, 37]]
72 42
60 63
12 50
24 54
```

Save

Run

☐ Run with a custom test case (Input/Output)