



## Max Door Number Sum

There is a town in which H houses are situated in each of the S streets. Each house has a door number which is an positive integer. So these S\*H door numbers form a matrix.

From any house you can navigate to the house in the east(right) or to the house located south (bottom). Thus the navigation ends when there is no house in the east or in the south.

The program must print the maximum possible sum of the door numbers traversed during the above mentioned navigation.

### Input Format:

The first line will S and H separated by a space.

Next S lines will each contain H positive integer values (representing the door numbers) separated by a space.

### Output Format:

The first line will contain the maximum possible sum of the door numbers traversed.

### Boundary Conditions:

2 <= S,H <= 500

1 <= DoorNumber <= 10000

### Example Input/Output 1:

Input:

3 3

15 25 30

45 25 60

70 75 10

Output:

215

Explanation:

The maximum possible sum of 215 is obtained when we follow 15->45->70->75->10

### Example Input/Output 2:

Input:

4 4

4 10 10 4

4 9 9 4

2 2 10 1

9 9 2 7

Output:

52

Explanation:

The maximum possible sum of 52 is obtained when we follow 4->10->10->9->10->2->7

**Max Execution Time Limit: 10000 millisecs**

Ambiance

Java ( 12.0)



```

1▼ import java.util.*;
2▼ public class Hello {
3
4     public static long max = 0;
5     static long maxOper = 1;
6
7▼     public static void main(String[] args) {
8
9         Scanner sc = new Scanner(System.in);
10
11         int r = sc.nextInt();
12         int c = sc.nextInt();
13         int mat[][] = new int[r][c];
14
15         for(int i=0;i<r;i++) for(int j=0;j<c;j++) mat[i][j] = sc.nextInt();
16
17         for(int i=0;i<c;i++)
18             solveMatrix(mat,0,i,mat[0][i],1);
19
20         System.out.println(max+"");
21     }
22
23▼     public static void solveMatrix(int[][] mat,int cRow,int cCol,long cSum
        ,int cOper){
24
25         if(cOper>=maxOper && cSum<max) return;
26
27         max = Math.max(max,cSum);
28
29         if(cRow+1<mat.length)
30             solveMatrix(mat,cRow+1,cCol,cSum+mat[cRow+1][cCol],oper+1);
31
32         if(cCol+1<mat[0].length)
33             solveMatrix(mat,cRow,cCol+1,cSum+mat[cRow][cCol+1],oper+1);
34
35     }
36 }

```

1912067@nec

Please wait while we run the program ....

