Question **1**

Correct

Marked out of 10.00

Given a gold mine of m x n dimensions.

1 Each field in this mine contains a positive integer which is the amount of gold present at that position.

2 Initially the miner is at first column but can be at any row.

3 He can move only int three directions: right, right-up and right-down. That is, east, northeast and southeast from a given cell.

Starting from any row (but always the first column), find out the maximum amount of gold that the miner can collect.

Input format

The first line contains T, the number of test cases. The following T lines contain 2 integers m and n which tell us the rows and columns respectively. This is followed by all the 2D array elements in a single line, separated by space.

Output format

For each test case, output the maximum number of gold collected.

Example Input #1

1

4 4 1 3 1 5 2 2 4 1 5 0 2 3 0 6 1 2

Output #1

16

Explanation:

(2,0) -> (1,1) -> (1,2) -> (0,3) OR (2,0) -> (3,1) -> (2,2) -> (2,3)

For example:

Input	Result
1 4 4 1 3 1 5 2 2 4 1 5 0 2 3 0 6 1 2	16

Answer: (penalty regime: 0 %)

```
t=int(input())
for i in range(t):
    inp=list(map(int,input().split()))
    m=inp[0]
    n=inp[1]
    arr=inp[2:]
    gold=[]
    idx=0
    for i in range(m):
        row=[]
        for j in range(n):
            row.append(arr[idx])
            idx+=1
        gold.append(row)
dp=[[0]*n for i in range(m)]
for col in range(n-1,-1,-1):
    for row in range(m):
        right=dp[row][col+1] if col!=n-1 else 0
        right_up=dp[row-1][col+1] if row>0 and col!=n-1 else 0
        right_down=dp[row+1][col+1] if row<m-1 and col!=n-1 else 0
        dp[row][col]=gold[row][col]+max(right,right_up,right_down)
max_gold=dp[0][0]
for i in range(1,m):
    if dp[i][0]>max_gold:
        max_gold=dp[i][0]
print(max_gold)
```

	Input	Expected	Got	
✔	1 4 4 1 3 1 5 2 2 4 1 5 0 2 3 0 6 1 2	16	16	✔

Passed all tests! ✔