Philosophers Stone

One of the secret chambers in Hogwarts is full of philosopher's stones. The floor of the chamber is covered by h × w square tiles, where there are h rows of tiles from front (first row) to back (last row) and w columns of tiles from left to right. Each tile has 1 to 100 stones on it. Harry has to grab as many philosopher's stones as possible, subject to the following restrictions:

- He starts by choosing any tile in the first row, and collects the philosopher's stones on that tile. Then, he moves to a tile in the next row, collects the philosopher's stones on the tile, and so on until he reaches the last row.
- When he moves from one tile to a tile in the next row, he can only move to the tile just below it or diagonally to the left or right.

Given the values of h and w, and the number of philosopher's stones on each tile, write a program to compute the maximum possible number of philosopher's stones Harry can grab in one single trip from the first row to the last row.

Input

The first line consists of a single integer T, the number of test cases. In each of the test cases, the first line has two integers. The first integer h (1 <= h <= 100) is the number of rows of tiles on the floor. The second integer w (1 <= w <= 100) is the number of columns of tiles on the floor. Next, there are h lines of inputs. The i-th line of these, specifies the number of philosopher's stones in each tile of the i-th row from the front. Each line has w integers, where each integer m (0 <= m <= 100) is the number of philosopher's stones on that tile. The integers are separated by a space character.

Output

The output should consist of T lines, (1 <= T <= 100), one for each test case. Each line consists of a single integer, which is the maximum possible number of philosopher's stones Harry can grab, in one single trip from the first row to the last row for the corresponding test case.

Example

Input:

```
1
6 5
3 1 7 4 2
2 1 3 1 1
1 2 2 1 8
2 2 1 5 3
2 1 4 4 4
5 2 7 5 1
```

Output:

```
32
```

//7+1+8+5+4+7=32