

# Aeroplane Bombing

You'll be given a grid as below:

```
0 1 0 2 0
0 2 2 2 1
0 2 1 1 1
1 0 1 0 0
0 0 1 2 2
1 1 0 0 1
x x S x x
```

In the grid above, 1: This cell has a coin. 2: This cell has an enemy. 0: It contains nothing. The highlighted(yellow) zone is the control zone. S is a spaceship that we need to control so that we can get maximum coins. Now, S's initial position will be at the center and we can only move it right or left by one cell or do not move. At each time, the non-highlighted part of the grid will move down by one unit. We can also use a bomb but only once. If we use that, all the enemies in the 5×5 region above the control zone will be killed. If we use a bomb at the very beginning, the grid will look like this:

```
0 1 0 2 0
0 0 0 0 1
0 0 1 1 1
1 0 1 0 0
0 0 1 0 0
1 1 0 0 1
x x S x x
```

As soon as, the spaceship encounters an enemy or the entire grid has come down, the game ends. For example, At the very first instance, if we want to collect a coin we should move left( coins=1). This is because when the grid comes down by 1 unit we have a coin on the second position and by moving left we can collect that coin. Next, we should move right to collect another coin( coins=2). After this, remain at the same position( coins=4). This is the current situation after collecting 4 coins.

0 1 0 2 0 0 1 0 0 0 0 2 2 2 1 -->after using 0 0 0 0 1 x x S x x -->bomb x x S x x Now, we can use the bomb to get out of this situation. After this, we can collect at most 1 coin. So maximum coins=5.

## Input Format

First line conatins t, number of test cases Next line contains N for each test Next N lines conatin input ( 5 integers )

## Constraints

$1 < t < 10$   $1 < N < 10$

## Output Format

For each test output single integer with format #test\_number output

## Sample Input 0

```
5
7
1 2 0 0 1
2 0 0 1 0
0 1 2 0 1
1 0 0 2 1
0 2 1 0 1
0 1 2 2 2
1 0 1 1 0
5
1 1 0 0 0
1 2 2 2 1
1 1 2 2 1
2 2 2 1 2
2 2 0 2 0
6
2 2 2 2 2
0 0 0 0 0
0 0 2 0 0
2 0 0 0 2
0 0 0 0 0
1 2 2 2 1
10
2 2 2 2 0
1 2 0 0 2
0 2 0 0 0
2 2 0 2 2
0 2 2 2 0
0 0 0 0 0
1 0 0 0 2
0 0 0 0 0
0 2 0 2 1
0 2 2 2 0
12
2 2 0 2 2
0 1 0 2 1
0 2 0 1 0
2 1 2 1 0
0 2 2 1 2
0 1 2 2 2
0 2 1 0 2
2 0 1 1 2
2 1 1 0 1
0 2 2 0 0
2 0 1 2 2
2 2 1 2 1
```

Sample Output 0

```
#1 6
#2 3
#3 0
#4 1
#5 9
```