

Your company has just constructed a new skyscraper, but you just noticed a terrible problem: there is only space to put one game room on each floor! The game rooms have not been furnished yet, so you can still decide which ones should be for table tennis and which ones should be for pool. There must be at least one game room of each type in the building.

Luckily, you know who will work where in this building (everyone has picked out offices). You know that there will be T_i table tennis players and P_i pool players on each floor. Our goal is to minimize the sum of distances for each employee to their nearest game room. The distance is the difference in floor numbers: 0 if an employee is on the same floor as a game room of their desired type, 1 if the nearest game room of the desired type is exactly one floor above or below the employee, and so on.

Input

The first line of the input gives the number of test cases, T ($1 \leq T \leq 100$). T test cases follow. Each test case begins with one line with an integer N ($2 \leq N \leq 4000$), the number of floors in the building. N lines follow, each consists of 2 integers, T_i and P_i ($1 \leq T_i, P_i \leq 10^9$), the number of table tennis and pool players on the i -th floor. The lines are given in increasing order of floor number, starting with floor 1 and going upward.

Output

For each test case, output one line containing ‘Case # x : y ’, where x is the test case number (starting from 1) and y is the minimal sum of distances.

Note: In the sample case, you can build a table tennis game room on the first floor and a pool game room on the second floor. In this case, the 5 pool players on the first floor will need to go one floor up, and the 4 table tennis players on the second floor will need to go one floor down. So the total distance is 9.

Sample Input

```
1
2
10 5
4 3
```

Sample Output

```
Case #1: 9
```