Dirty Plates

Lazy Louie loves to eat but he adamantly hates cleaning. That is why he has chosen to delay cleaning for as long as possible. Being an avid inventor, Louie created an invention to help increase his laziness. That invention is the two sided plate! Two sided plates can be stacked like coins for easy storage. They also have the interesting feature that you may eat from either side of the plate.

Lazy Louie lacks cabinet space. That is why he stores his two sided plates directly on his dining table. When enjoying a meal he simply eats off the topmost plate, although he will only eat off the topmost surface if it is clean. When the plates are stacked, if a dirty side of the plate touches the clean side of another plate then the grime of that plate transfers to the clean plate making that side of the plate dirty. The side that was originally dirty stays dirty.

Since Louie is stacking plates on his table, if a dirty side of a plate touches the table then the grime will transfer to the table making the table dirty. If the table is dirty already and it touches a clean side of a plate then that plate's side becomes dirty. Note that the table remains dirty. Also, both sides remain dirty when two dirty sides, either the table or plates, touch.

The Problem:

Louie would like to know the maximum number of meals that can be eaten before any cleaning is done. Louie is given the number of plates he already has of three kinds: plates that are clean on both sides, plates that are clean on one side while dirty on the other, and plates that are dirty on both sides. Before eating his first meal he can stack these plates in any way he likes on the table. After eating each meal the topmost plate becomes dirty. Between meals Louie can rearrange the plates in any way he likes. Louie is allowed to change the order of the plates on

the stack and change which side is facing up but they must remain a stack of plates after being

rearranged.

Input

The first input line contains a positive integer, n, indicating the number of eating scenarios to

analyze. The next n lines contain the description of the plates. Each line contains three integers,

c, s, and d, (0 ≤ c ≤ 100; 0 ≤ s ≤ 100; 0 ≤ d ≤ 100) representing (respectively) the number of

plates with both sides clean, the number of plates with one side clean and the number of

completely dirty plates.

Output

For each scenario, first output the heading "Scenario #d: ", where d is the scenario number,

starting with 1. Then, print the maximum number of times a meal can be eaten before Louie has

to clean. Follow the format illustrated in Sample Output.

Example

Input:
```
4
1 0 0
2 0 0
1 1 3
2 2 2
```

Output:
```
Scenario #1: 2
Scenario #2: 3
Scenario #3: 2
Scenario #4: 4
```