

Beggars

Begging nowadays is an organized profession and team work is extremely important. Beggars target special occasions to maximize revenue, such as Muslim Friday prayers. They wait outside the mosque until the prayer ends and ask people for money as people leave the mosque. In this problem you are to help two beggars to maximize their revenue on Friday after the Friday prayers. These beggars are experienced and know everything they need, specifically they know exactly when each mosque ends the prayer and how much money they will gain if at least one of them stands in front of that mosque when the prayer ends (there is no use if both of them stand at the same mosque).

In their town there are n mosques on a straight line and you will be given the x coordinate for each mosque. The time needed for a beggar to travel from one mosque a to another mosque b is $|x_a - x_b|$ units of time. As you know, these baggers are professionals and take no time to collect the money from a mosque if they happen to be there when the prayer ends, and can immediately start moving to another mosque.

Of course the beggars can choose to initially start from any mosque. Your task here is to compute the maximum amount of money they can collect together.

Input Specification

Input contains multiple test cases. Each test case starts with number of mosques $1 \leq n \leq 100$, followed by n lines. Each line consists of three 32-bits signed integers x , t , and m representing the x -coordinate of the mosque, the time when the prayer ends, and the amount of money that can be collected from this mosque. The input will be terminated when n equals 0 and should not be treated as a test case.

Output Specification

For each test case you should print a single integer, the maximum amount of money that can be collected by the two beggars.

Sample Input

```
3
7 6 19
2 3 18
9 8 13
4
1 4 5
3 4 5
2 5 5
4 5 5
4
1 4 5
3 4 5
```

2 5 5

5 5 5

0

Sample Output

50

20

15