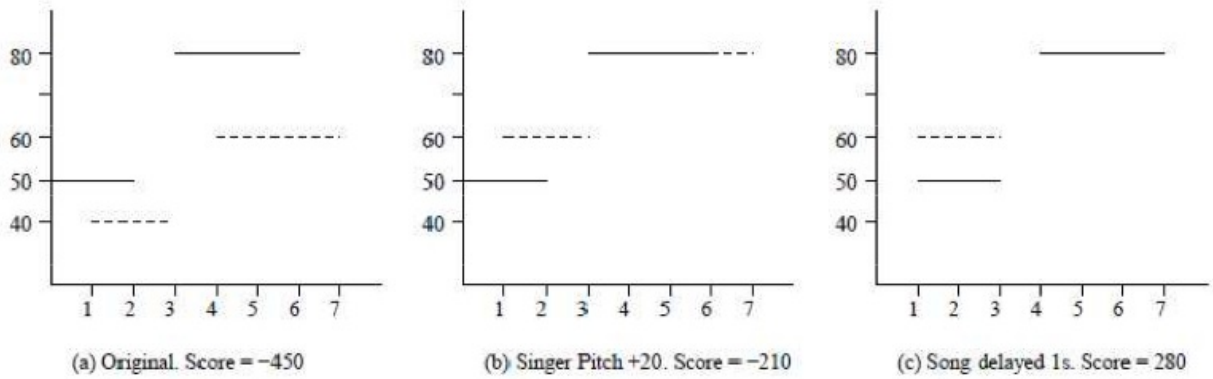
C - Karaoke

“But I sing awfully!” - Gonzo kept saying. “Come on! Nobody cares, it just for fun!” – was the common reply of his friends. “And we’ll give you all our support!”. But Gonzo was not convinced. Although he was an accomplished shower singer, he didn’t like to do it in public. He had played karaoke games before and his problems were always the same: he could pronounce the words and keep a note very well, but typically it was not the right one, and not at the correct time. That’s why in these games his scores were always poor.

But there was something in his pocket to save the day. He had a friend who just came from a programming contest and had given him a pendrive with the solution: A Corrector for Music (ACM), a program that can adjust the notes that the game detects, so that he could achieve the a better score. Gonzo was relieved. For once he wouldn’t be the worst singer of the group. Now he had just to use the program properly. The karaoke game displays a screen with several horizontal bars, representing words of the song, at different heights over a time axis. The y-coordinate of each bar indicates the *pitch* of that word. This is, the note that the game expects to detect from the singer. The game receives the voice of the singer from the microphone, and detects the pitch and the time interval where the note was detected. By comparison with the original song, the game displays a new set of horizontal bars in the same graph of the originals, and calculates the score according to the following rules:

- The singer starts the game with 0 points.
- If the song pitch and the singer pitch coincides, the singer wins 100 points by each second that the coincidence continues.
- If the song pitch and the singer pitch differ, the singer loses the absolute difference between the pitches each second that such difference exists.
- If there is no song pitch, but the singer sings; or inversely, if there is a song pitch, but the singer doesn’t make a sound, the singer loses 100 points each second that this happens.
- While there is no song pitch, and no input from the microphone, the score remains the same.

ACM modifies the pitch of all the singer words by adding an integer to all of them. There is no possibility of modifying just an specific part of the singer’s notes. The pitch is always a positive number. If by any chance a pitch number drops to zero or less, the game interprets that no sound has been received. The program can also delay the time when the song begins or the time when the singer’s voice begins to be read, by adding an integer number of seconds in order to better adjust the intervals. The figure shows an example with the song words (solid lines) and the singer’s words (dashed lines).



You will receive the details of the song that is stored in the game, and the words detected by the microphone. With that information you must determine the maximum possible score that can be obtained by adjusting the pitch of the singer's words, and delaying the song or the voice.

Input

The input contains several test cases. The first line of a test case contains one integer N indicating the number of words that are included in the song (1<=N<=200). Each of the next N lines contain six integers s_{i1}, f_{i1}, p_{i1}, s_{i2}, f_{i2} and p_{i2}. The first three of those integers represent the continuous interval of time $[s_{i1}, f_{i1})$, where the word i must be sung with pitch pi1. The following three integers represent the continuous interval $[s_{i2}, f_{i2})$, where the word i of the singer is detected with pitch p_{i2}. For all cases, $0 < s_{i1} < f_{i1}$, $0 < s_{i2} < f_{i2}$, $20 < p_{i1}, p_{i2} < 120$. The end of input is indicated by a line containing only one zero.

Output

For each test case in the input, your program must print a single line, containing the maximum score that can be achieved in the game by using the ACM program.

Example

Input:

```
1
1 4 60 6 9 70
1
1 4 60 6 8 70
2
0 2 50 1 3 40
3 6 80 4 7 60
0
```

Output:

```
300
100
280
```