

### Task 1:

After taking the inputs correctly I sorted the input-output pair based on the end timing of the task using lambda function. After that I took an empty list to store the output and took a variable 'end' which initialized by 0, it is for to compare the end time of the tasks. Then I iterated the sorted list of each pair, here I used conditional statement to check whether the current end time is less than or equal to current task. If it is then I append that task into res list and lastly updated the end variable to that task end time. This loops continue for every task till the end. And finally I printed the len of the res task then added those selected task in the output file.

### Task 2:

Here the input processing part is same as task 1. Then I took two variable, complete\_task initialized with 0 of size m, it will keep track of time when each person finishes their assigned task and assigned\_task=0, it is for the track of the number of tasks that has been assigned. Then I have iterated through the input sorted list I using task variable. Here firstly I set false in assigned variable. Then in the second for loop if for one person any task has not been assigned yet and any tasks start time is greater or equal than the task the person has completed before then it indicates that person is applicable for this task. Then I set the assigned variable as true. Updated the completed\_task variable to the current working task ending time. And lastly incremented by 1 in assigned\_tasks variable. And lastly used break because the current work is done by the person. This process will go in till the end. Lastly writes the total number of task assigned to people into the output file as string.

### Task 3:

Firstly I took each friendship pair in a nested list. Then I took a nested list variable "x" where I initialized each list from 1 to n to keep the friend circles. And an empty list to store the sizes of friend circles after each query. Then I am iterating the input list. Here first I have unpack a,b from each nested list and took another two variable m=-1 & n=-1, to keep the indexing track of a and b respectively. The next for loop iterates through x list and the loop index is j. Inside the there is two if condition to find out correct index of a and b. Like if a in x[j] then it will update the index for a in m same goes for b. Then if m and n is not same it means they are not friend yet so I am adding them as one, then deleting their individual list from the x list. Lastly I append them inside x list, and their combined length in s list. And if the m and n are same it means they are already in the same friend circles so here I have append the friend circle list length in s list. This process will go on for pair of the input list. And lastly I have added those friend circles list length in the output file.