First Time Stuff

So, before giving you all the first assignment let us get some things clear. Every assignment would have a deadline which will be around one week (maybe more).
Our aim is not only to teach you algorithms but also try to make you understand their real-life applications so we will try to create problems based on some real-life stories or things we could imagine.

This whole project would be a journey where you and us, we all would learn many things.
We are here because of our passion for algorithms.
One thing I want to be clear about is that we got 200 applications and you guys are among the selected 17 so, please avoid "FRAUD", as there are many boys/girls who we couldn't select because of limited seats.

I hope you will learn many things during this period other than algorithms too!
So why not begin with git?

Learn about "Version Control".

Why?

Assignment Submission (Current Plan)

- Make a git repository named "Algorithms InDepth".
- Keep it private (Until we allow to make it public (the reason is obvious))
- Add we both as collaborators. (We will send you our usernames)
- In it create a separate directory for each assignment.
- For each assignment directory create two directories, one named "Problem" and other named "Solution"
- And put your stuff accordingly.
- Make timely commits with up to the point commit description!

Any doubts? Feel free to ask in our messenger group.

So, let us begin with our first assignment which is based on the stuff you have learnt in your ESC101, but most of you must have ignored it.

Assignment 1 Deadline - 27/05/2019

PROBLEM: ***Weighted Event Scheduling (Part 1)***

You are the Software Engineer currently employed by ALPHABET Inc. (Don't know what is it? Google it buddy!).
As you must know by now is that Alphabet has many companies under it including Google, DeepMind, Fibre, etc. There is a small problem which they are facing. As we all know these big companies have a systematic structure for their event scheduling. But we will try to make a smaller version for the same.
Being such mammoth organizations, they all have some meetings and events scheduled beforehand. But they can only participate or organize those events which are more important to them. So they gave you the task to help them with that.

You have planned the structure of your application but the main thing is the algorithm to solve this problem. The first thing that came to your mind is that let's sort the events based on their priorities. So, let's do this!

Sorting seems to be an easy task! Isn't it!?
Dude, we learnt it in our first year of our college!

Aha! But there is a catch, We mostly deal with idealistic problems. But in the real world, nothing is ideal. And being a Good Programmer you should write an algorithm which is effective in its *Time as well as Space Complexity*. Hardware is too costly dude!

And since these organizations are huge you will surely get large amounts of data. If you plan to store the data in an array and apply Merge Sort, I will urge you to think upon it once more!

Your application will get data dynamically and you will never know the amount of data you are gonna get so what's the solution?

Dynamic Memory Allocation

Yeah! You heard it right. You are gonna have fun with pointers in this assignment and believe me they are the most important stuff for being an awesome programmer.

And since you are writing code for the server and your application is helping all of your subsidiaries simultaneously, your server is gonna get data from all. The companies send you the data as and when they have an event planned. So your data will be arranged in the order you receive them and not according to the companies!

Your application will receive data from all your subsidiary companies in the form of

X S-E Y

Where X is the ID of the company which sent you the data. (Assume $1 \leq X \leq N$, where N is the number of different companies which send you data)
S is the start time of the event. And E the end time. (there is a '-' in between them!)
Format of both - HH: MM (24-hour clock format)
Y is the importance of the event. It will be a real number calculated by some algorithm which the organization must have chosen (No need to care about that algorithm right now).

Since you are getting data dynamically, you should store data in a dynamic way!
What is the best choice based on current knowledge?

LINKED LIST OF STRUCTURES

So, store data in a linked list (MAKE SEPARATE LINKED LIST FOR EACH ORGANISATION).

Now, we are to sort the data for each organization based on the Y (importance).
So, how we will fulfill our target?

We will write a Merge Sort!
But!!!
Since we are Good Programmers, we don't want to waste memory.
So, as we all must have written the merge sort in array and in that, in the merge function we used to create two separate arrays and copy data and then destroy those arrays!?

But in our Linked List implementation we have the power of "Creating and Breaking Links", so we'll not create any other linked list or array in our merge function. Instead, we'll rearrange our current linked list only by breaking and creating links!

Finally, output your data as

X Y1 Y2 Y3 . . .

where, X is a company's ID and Yi's are there event's importance arranged in increasing order. (upto 3 decimal digits precision)
Also, output companies data in increasing order of their IDs

Example:

INPUT	OUTPUT
1 12:30-14:45 50.23 3 11:00-12:00 21.232 2 01:12-12:32 123.0 2 12:12-12:13 1.12 1 08:08-12:59 242.123 1 12:45-13:15 12.123	1 12.123 50.230 242.123 2 1.12 123.000 3 21.232

Rules for Writing Codes

Firstly, for this assignment, you are language bound.

YOU CAN ONLY USE "C" (not even C++) IN THIS ASSIGNMENT

You have to follow the professional style of coding

- Give Variable names that describe the variable's objective (same for function)
- Proper Indentation (Don't know how? Google it!)
- Use functional implementation, i.e. make relevant functions to do the subtask.
- Proper Commenting is a must!

Problem Credits: Aniket Sanghi