Problem set-1

Task A

Problem:
When users post an update on social media,such as a URL, image, status update etc., other users in their network are able to view this new post on their news feed. Users can also see exactly when the post was published, i.e, how many hours, minutes or seconds ago.
Since sometimes posts are published and viewed in different time zones, this can be confusing. You are given two timestamps of one such post that a user can see on his newsfeed in the following format:
Day dd Mon yyyy hh:mm:ss +xxxx
Here +xxxx represents the time zone. Your task is to print the absolute difference (in seconds) between them.

Input Format:
The first line contains T, the number of test cases.
Each test case contains 2 lines, representing time t1 and time t2.

Constraints:

1. Input contains only valid timestamps
2. year<=3000

Output Format:
Print the absolute difference (t1-t2) in seconds.

Sample Input:

```
2
Sun 10 May 2015 13:54:36 -0700
Sun 10 May 2015 13:54:36 -0000
Sat 02 May 2015 19:54:36 +0530
Fri 01 May 2015 13:54:36 -0000
```

Sample Output:

```
25200
88200
```

Explanation:
In the first query, when we compare the time in UTC for both the time stamps, we see a difference of 7 hours. which is 7x3600 seconds or 25200 seconds.
Similarly, in the second query, the time difference is 5 hours and 30 minutes for the time zone adjusted so that we have a difference of 1 day and 30 minutes. Or 24x3600+30x60 => 88200

Y*ou can use rust/python to accomplish this task.*

Task B

Build a rust/python service which will take Task A as an input(plain text with new line) through rust/python rest api. Output will be the same as Task A as response. Response will be an ordered json array with the result.

Example: ["25200", "88200"]

Please Note that, you have to provide a readme file with the process of running this application.

Task C

Make your app containerize(docker) and run at least two images so that any app instance can provide the result as a response. Also add node id from which node it is serving.

Example:

```
{
  "id":"YOUR_NODE_ID",
  "result":[ "25200", "88200"]
}
```

Please Note that, you have to provide a readme file with the process of running this application.
Commit all your tasks in git repo and share the link with us.

Problem set-2

Audio and Video call calculation from IPDR(attached in excel sheet)

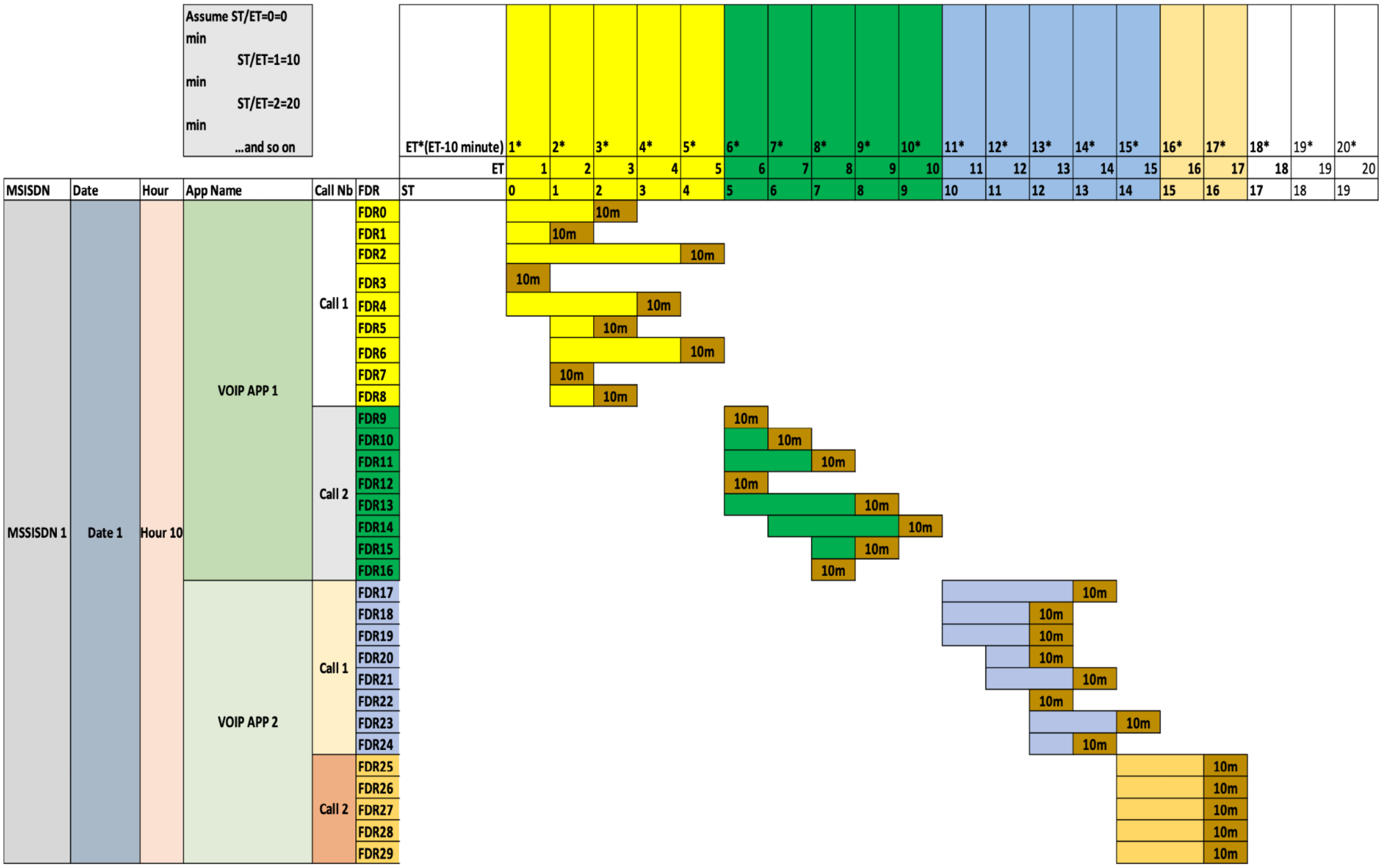
1. Select each MSISDN.
2. Select Specific start and end datetime domain/app wise.
3. For each VoIP APP, identify each single call as like the **graphical table** at the bottom of this document
    a. Need to calculate first ST (start time), ET (End Time) for each FDR.
    b. then to calculate ET*(ET-10 min) for each FDR to exclude idle time (10 min) of each
4. FDR
    a. If ET-10 min < ST then keep the original ET.
5. Calculate Total volume of each call of each domain=(Sum of DL volume of all FDR + Sum of UL volume of all FDR) in Kb. Value of UL and DL in CDR is in Byte.
6. Calculate Total time of each call of each VoIP App= [each call Highest ET* among respected FDR's(minute) - each call lowest ST among respected FDR's (minute)] in sec.
7. Calculate bit rate(kbps) of each call of each VoIP App= (Total volume of each call of each VOIP App/Total time of each call of each VOIP App)
8. Identification of Audio or video call and its count,
    a. For each VoIP APP, if bit rate(kbps) of each call of each VoIP App <10kbps, discard the call record.
    b. Assuming <=200 Kbps is audio call,>200 kbps is video call

Output:

Msisdn, domain, durations in sec, fdr_count(number of CDR to make a single call), kbps, isAudio, isVideo

- **Please solve the problem in your preferred processing framework/tools/programming and commit all your tasks in git repo and share the link with us.**
- **Please Note that, you have to provide a readme file with the process of how you have done it, architectural diagrams, etc.**
- **Please also note that problem set-2(this problem) carries double the mark than problem set-1.**

Graphical table:



			Assume ST/ET=0=0 min ST/ET=1=10 min ST/ET=2=20 min ...and so on			ET*(ET-10 minute)	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*	11*	12*	13*	14*	15*	16*	17*	18*	19*	20*
						ET	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MSISDN	Date	Hour	App Name	Call Nb	FDR	ST	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
MSSISDN 1	Date 1	Hour 10	VOIP APP 1	Call 1	FDR0		■	■	10m																	
					FDR1		■	10m																		
					FDR2		■	■	■	■	10m															
					FDR3		10m																			
					FDR4		■	■	■	10m																
					FDR5			■	10m																	
					FDR6			■	■	■	10m															
					FDR7			10m																		
					FDR8			■	10m																	
				Call 2	FDR9							10m														
					FDR10							■	10m													
					FDR11							■	■	10m												
					FDR12							10m														
					FDR13							■	■	■	10m											
					FDR14								■	■	■	10m										
					FDR15									■	10m											
					FDR16									10m												
			VOIP APP 2	Call 1	FDR17												■	■	■	10m						
					FDR18												■	■	10m							
					FDR19												■	■	10m							
					FDR20													■	10m							
					FDR21													■	■	10m						
					FDR22														10m							
					FDR23														■	■	10m					
					FDR24														■	10m						
				Call 2	FDR25																■	■	10m			
					FDR26																■	■	10m			
					FDR27																■	■	10m			
					FDR28																■	■	10m			
					FDR29																■	■	10m			