**Problem 1:**

Write a function babylonian\_sqrt(num) that calculates square root of a number using the Babylonian Method.

Note: You can only use DMAS operations/functions, power operator i.e. \*\* , and round() function in Python, and you cannot use any other modules/packages for square-root calculation.

Useful Links:

* [Babylonian Method for Square Root Algorithm](https://blogs.sas.com/content/iml/2016/05/16/babylonian-square-roots.html)

**Problem 2:**

You are given a list of emails of events; examples of such emails (defined as a list of Python strings) are below:

emails = [

'''

Sender: person1@site29.com

CC: ccperson1@site26.com, ccperson.32@site0.net, cc.person5@udacity.org

BCC: bccperson43@site99.com, bccperson.42@site21.net.uk, bcc.person.7@google.org.pk

Body:

We are happy to invite you to the following events:

Raj's birthday at 29-10-2023 on 06:00 PM.

Sheila's morning yoga practice dated 2022-03-31, 09:30 AM

'''

,

'''

Sender: person1@site59.net.pk

CC: ccperson1@site26.com, ccperson.32@site0.net, cc.person5@udacity.org

BCC: bccperson43@site99.com, bccperson.42@site21.net.uk, bcc.person.7@google.org.pk

Body:

You are cordially invited to:

Avril's business meeting on Environmental Disasters, airing live on 2023-02-08 at 01:00 PM.

Sheila's yoga practice on 12:30 PM, 31-03-2022

'''

]

Now, here's the general format of a particular email, line by line:

Line # 1: Sender: <sender\_email>

Line # 2: CC: <cc\_email1>, <cc\_email2>, <cc\_email3>, …

Line # 3: BCC: <bcc\_email1>, <bcc\_email2>, <bcc\_email3>, …

Line # 4: Body:

Line # 5: <greeting>

Line # 6: <event1>

Line # 7: <event2>

.

.

.

Last Line: <eventN>

Your task is to write a Python function named extractFromEmails(emails), that extracts senders, CCs, BCCs, and events' date and time, from all emails, and stores them into separate lists, like so (using above given emails):

senders = [ 'person1@site29.com', 'person1@site59.net.pk' ]

CCs = [

[ 'ccperson1@site26.com', 'ccperson.32@site0.net', 'cc.person5@udacity.org' ], # CCs from emails[0]

[ 'ccperson1@site26.com', 'ccperson.32@site0.net', 'cc.person5@udacity.org' ] # CCs from emails[1]

]

BCCs = [

[ 'bccperson43@site99.com', 'bccperson.42@site21.net.uk', 'bcc.person.7@google.org.pk’ ], # BCCs from emails[0]

[ 'bccperson43@site99.com', 'bccperson.42@site21.net.uk', 'bcc.person.7@google.org.pk' ] # BCCs from emails[1]

]

events\_details = [

[ ['29-10-2023', '06:00 PM' ], [ '2022-03-31', '09:30 AM' ] ], # events' details in emails[0]

[ [ '2023-02-08', '01:00 PM' ], [ '31-03-2022', '12:30 PM' ] ] # events' details in emails[1]

] # note that order of event details is consistent irrespective of its occurrence in the email i.e. date is always first, and then time of event is stored in list.

PS: Each string, be it sender, CC, BCC, or event details, should strictly contain only the relevant characters i.e. any leading/trailing spaces, punctuation marks etc are to be removed before returning from extractFromEmails(emails) function, so, for example, an event time of '06:20 PM.' should be '06:20 PM' i.e. no period/full-stop at the end; 'ccperson1@site26.com,' should be 'ccperson1@site26.com' i.e. no comma at the end; ' ccperson.32@site0.net ' should be 'ccperson.32@site0.net' as there should be no leading or trailing spaces/tabs. And emails should also be captured by regular expressions of emails, instead of just comma separated values.

Allowed functions from module 're':

* re.find(…)
* re.findall(…)
* re.sub(…)
* re.split(…)

Useful Links:

* ['re' module and regular expression documentation](https://docs.python.org/3/library/re.html)