

Given a file, parse through all lines in the file and extract all hostnames

from valid email addresses. Then, print sorted the list of all unique hostnames found in ##### the file. example output: ['brunswickschool.org', 'd.tv',]

follow method decomposition by first extracting email from a line of text

write COMMENTS ¶

''' Given a line of text, if it contains an email address, return the hostname. Each line contains at most one email address, but a line may not contain an email address, in which case you should return an empty string: ''. An email address is defined as: username@host.name, where the hostname is made of alphabetic chars and periods. To be valid, a hostname must be at least length 4, and contain at least 1 period. example input: extractHostname('Please forward this email to ali@brunswickschool.org. Thanks!') example output: 'brunswickschool.org'

```
extract the FIRST LINE OF TEXT FROM THE TEXT file and use it as a parameter
for this function.
'''
```

```
In [26]: def readFile(filename):
          #Reads file
          with open (filename) as fopen:
              myList = fopen.read().replace('\n', ' ').split(' ')
          return myList
          print(readFile('emails.txt'))

['Please', 'forward', 'this', 'email', 'to', 'josh@brunswickschool.or
g', 'for', 'me.', 'Thanks!', 'Can', 'someone', 'tell', 'me', 'who', 'ow
ns', 'the', 'jess@yahoo.com', 'email', 'address?', 'The', 'email', 'gup
ta@gmail.com', 'keeps', 'sending', 'me', 'spam', 'mail.', 'Please', 'fo
rward', 'this', 'email', 'to', 'tommy@brunswick.edu', 'for', 'me.', 'Th
anks!', 'Fran@ye', 'and', 'chef@me', 'is', 'my', 'favorite!', 'Do', 'yo
u', 'think', 'ali@b.c', 'is', 'spam?', 'This', 'one', "isn't", 'spam:',
'jack@d.tv', 'Hello,', 'world!', 'Why', 'am', 'I', 'getting', 'emails',
'from', 'andrew@spam.com?']
```

```
In [23]: def extractHostname(filename, line):  
    #Finds the first host name in the line you specified  
    hostnames=[]  
    with open (filename) as fopen:  
        myList = fopen.read().split('\n')  
    #Isolates the wanted line  
    finalList=myList[line].split(' ')  
    #Finds the first hostname and returns it  
    for x in finalList:  
        if x.find('@') !=-1:  
            return x.split('@')[0]  
    #Default return if no hostname  
    return hostnames  
print(extractHostname('emails.txt',2))
```

gupta

```
In [24]: def extractAllHostnames(filename):  
         #returns all host names  
         file = readFile(filename)  
         hostnames=[]  
         #Searches for host names  
         for x in file:  
             if x.find('@')!=-1:  
                 #Adds hostnames to list  
                 hostnames.append(x.split('@')[0])  
         return hostnames  
extractAllHostnames('big-emails.txt')
```

```
Out[24]: ['in',
          "alice's",
          'fulcrum',
          'the',
          'of',
          'she',
          'reading,',
          'conversations',
          '`without',
          'mind',
          'her',
          'would',
          'and',
          'eyes',
          'so',
          'it',
          'itself,',
          'it',
          'the',
          'actually',
          'started',
          'it',
          'or',
          'it,',
          'field',
          'large',
          'another',
          'in',
          'a',
          'then',
          'to',
          'very',
          'deep,',
          'plenty',
          'next.',
          'was',
          'she',
          'filled',
          'maps',
          'from',
          "marmalade'",
          'jar',
          'of',
          'past',
          'such',
          'tumbling',
          'me',
          'very',
          'end!',
          'many',
          'the',
          '"',
          'had',
          'in',
          'showing',
          'to',
          'right',
```

'latitude',
'grand',
'presently',
'through',
'the',
'there',
'all',
'have',
'new',
'she',
'air!',
'asking!',
'never',
'down',
'me',
'was',
'of',
'in',
'catch',
'eat',
'sleepy',
'sometimes',
"cats?",
'matter',
'off',
'saying',
'you',
'came',
'and',
'alice',
'before',
'was',
'down',
'alice',
'ears',
"it's",
'but',
'row',
'from',
'doors',
'side',
'trying',
'wondering',
'three-legged',
'except',
'of',
'doors',
'any',
'a',
'it',
'she',
'great',
'and',
'she',
'looked',
'dark',
'beds',

```
'head',  
'`it',  
'little',  
'like',  
'to',  
'that',  
'were',  
'no',  
'table,',  
'rate',  
'time',  
'round',  
'words',  
'printed',  
'`drink',  
'alice',  
'`and',  
'she',  
'had',  
'because',  
'red-hot',  
'and',  
'very',  
'that,',  
'with',  
'alice',  
'nice,',  
'flavour',  
'toast,)',  
'*',  
'*',  
'*',  
'a',  
'like',  
'high,',  
'the',  
'for',  
'however,',  
'felt',  
'this;',  
'a',  
'like',  
'not',  
'a',  
'more',  
'garden',  
'when',  
'when',  
'she',  
'and',  
'up',  
'had',  
'poor',  
'that!'",  
'advise',  
'very',  
'it),',
```

'bring',
'her',
'of',
'child',
'two',
'left',
'little',
'and',
'`eat',
"i'll",
'and',
'into',
'which',
'bit,',
'of',
'she',
'size:',
'much',
'expecting',
'stupid',
'off',
'*',
'*',
'*',
'pool',
'so',
'for',
'telescope',
'down',
'out',
'i',
'you',
'far',
'must',
'`or',
'go!',
'every',
'planning',
"carrier,'" ,
'seem,',
'will',
'right',
'the',
'(with',
'what',
'against',
'high,',
'golden',
'it',
'to',
'hopeless',
'ought',
'like',
"you!'",
'of',
'round',
'in',

'eyes',
'returning',
'gloves',
'in',
'as',
'so',
'was',
'her',
'the',
'gloves',
'could',
'gloves',
'fanning',
'how',
'went',
'think:',
'feeling',
'next',
"that's",
'that',
'for',
'she',
"doesn't",
'mabel',
'and',
'if',
'times',
'oh',
'shall',
'multiplication',
'london',
'certain!',
'for',
'lap',
'repeat',
'to',
'doth',
'improve',
'pour',
'on',
'he',
'neatly',
'and',
'with',
'are',
'with',
'poky',
'oh!',
'it;',
'down',
'again',
'first',
'up:',
'here',
'of',
'being',
'she',

'on',
'`how',
'small',
'and',
'was',
'rapidly:',
'dropped',
'shrinking',
'frightened',
'the',
"garden!'",
'alas!',
'lying',
'things',
'so',
'that',
'slipped,',
'was',
'she',
'case',
'had',
'you',
'children',
'spades,',
'them',
'was',
"hadn't",
'punished',
'in',
'queer',
'splashing',
'way',
'or',
'now,',
'that',
'alice,',
'down',
'at',
'way',
'tired',
'the',
'she',
'latin',
'a',
'seemed',
'said',
'alice;',
'william',
'alice',
'anything',
"chatte?'",
'french',
'water,',
'that',
"animal's",
'in',
'you',

'alice',
'about',
'if',
'to',
'sits',
'nicely',
'such',
'alice',
'mouse',
'talk',
"you'd",
'who',
'`as',
'low',
'the',
'a',
'of',
'eagerly:',
'show',
'such',
'hair!',
'beg',
'to',
'a',
'--',
'sorrowful',
'mouse',
'in',
'softly',
'and',
'round',
'pale',
'and',
'my',
'it',
'pool',
'it:',
'a',
'the',
'a',
'were',
'--',
'them',
'of',
'this',
'familiarily',
'life.',
'long',
'and',
'how',
'lory',
'be',
'who',
'out',
"enough!'",
'in',
'would',

'very',
'said',
'all',
'cause',
'to',
'late',
'northumbria',
'with',
'the',
'`did',
'said',
'proceed.',
'of',
'the',
'--',
'`found',
"it,"",
"it"",
'when',
'generally',
'the',
'but',
'go',
'offer',
'the',
'getting',
'melancholy',
'at',
'rising',
'the',
'energetic',
'`i',
'half',
'hide',
'tittered',
'offended',
'thing',
'she',
'had',
'no',
'`the',
'the',
'i',
'race-course',
'it',
'placed',
"away,"",
'not',
'running',
'dodo',
'is',
'and',
'not',
'a',
'(the',
'in',
'has',

'chorus',
'`why',',',
'crowded',
'out',
'put',
'the',
'got',
'all',
'have',
'very',
'went',
"thimble,'" ,
"here,'" ,
'round',
'your',
'elegant',
'all',
'all',
'could',
'think',
'looking',
'eat',
'large',
'patted',
'was',
'something',
'history',',',
'added',
'afraid',
'said',
'alice',',',
'said',
'it',
'kept',
'tale',
'said',
'that',
'met',
'both',
'law:',
'--',
"i'll",
'have',
'mouse',
'alice',
'you',
"pardon,'" ,
'fifth',
'had',
"knot!'" ,
'`oh',',',
'mouse',',',
'`you',
'alice.',',
'you',
'only',
'alice',

'others',
'and',
"wouldn't",
'opportunity',
'lesson',
"temper!",
'snappishly.',
'of',
'here,',
'particular.',
'might',
'the',
'replied',
'capital',
'wish',
'as',
'a',
'old',
'itself',
'getting',
'voice',
'dears!',
'was',
'to',
'seems',
'best',
'see',
'and',
'heard',
'in',
'his',
'finish',
'a',
'slowly',
'anxiously',
'heard',
'oh',
'ferrets!',
"wonder?",
'the',
'hunting',
'nowhere',
'with',
'little',
'very',
'what',
'a',
'fan!',
'in',
'mistake',
'to',
'out',
'gloves',
'came',
'with',
'went',
'fear',

'found',
'`to',
'for',
'she',
'and',
'your',
"think,'" ,
'house',
'people',
'found',
'table',
'of',
'just',
'little',
'label',
'it',
'to',
"i'll",
'i',
'tiny',
'indeed,',
'head',
'from',
'saying',
'any',
'i',
'went',
'kneel',
'even',
'effect',
'arm',
'window,',
'the',
'what',
'the',
'it',
'her',
'room',
'was',
'ordered',
'and',
'--',
'of',
'kind',
'the',
'book',
'one',
'grow',
'any',
'be',
'lessons',
'foolish',
'why,',
'all',
'the',
'but',
'stopped',

'my',
'then',
'the',
'the',
'the',
'afraid',
'tried',
'opened',
'heard',
'round',
'alice',
'the',
'snatch',
'hold',
'which',
'had',
'of',
'where',
'and',
'digging',
'rabbit',
'help',
'in',
'pronounced',
'why',
'fills',
'arm',
'any',
'long',
'such',
'at',
'last',
'little',
'of',
"they'll",
"i'm",
"don't",
'waited',
'the',
'all',
'ladder?',
'--',
'no',
'--',
'bear?',
'loose',
'was',
'do',
'bill!',
'the',
'to',
'to',
'be',
'but',
'foot',
'till',
'and',

'above',
'would',
'she',
'you',
'confusion',
'was',
'you?',
'last',
'more',
'i'm',
'comes',
'a',
'the',
'burn',
'`if',
'dinah',
'alice',
'herself',
"they'd",
'say',
'barrowful',
'not',
'little',
'her',
"i'll",
'better',
'noticed',
'into',
'one',
'my',
'it',
'cakes',
'she',
'the',
'quite',
'the',
'giving',
'the',
'she',
'herself',
'`is',
'is',
'i',
'very',
'simply',
'about',
'anxiously',
'a',
'puppy',
'one',
'she',
'all',
'hungry',
'spite',
'hardly',
'puppy',
'feet',


```
'believe',  
'keep',  
'appeared',  
'stick',  
'it',  
'and',  
'under',  
'began',  
'time',  
'back',  
'good',  
'half',  
'to',  
'was',  
'breath',  
'dear',  
'leant',  
'liked',  
'do',  
'i'd',  
'eat',  
'something']
```

```
In [13]: def extractDomains(filename, line):  
    #Finds the first host name in the line you specified  
    hostnames=[]  
    with open (filename) as fopen:  
        myList = fopen.read().split('\n')  
    #Isolates the wanted line  
    finalList=myList[line].split(' ')  
    #Finds the first hostname and returns it  
    for x in finalList:  
        if x.find('@') !=-1:  
            return x.split('@')[1]  
    #Default return if no hostname  
    return hostnames  
print(extractDomains('emails.txt',0))
```

brunswickschool.org

```
In [14]: def extractAllDomains(filename):  
    #Will make a dictionary with all the domains and how common they are  
    dict={}  
    file = readFile(filename)  
    domainnames=[]  
    #Finds the domains and puts them into a list  
    for x in file:  
        if x.find('@')!=-1:  
            domainnames.append(x.split('@')[1])  
    #Add the domains from the list into dict  
    for x in domainnames:  
        if x in dict:  
            dict[x]+=1  
        else:  
            dict[x]=1  
    return dict  
extractAllDomains('emails.txt')
```

```
Out[14]: {'brunswickschool.org': 1,  
          'yahoo.com': 1,  
          'gmail.com': 1,  
          'brunswick.edu': 1,  
          'ye': 1,  
          'me': 1,  
          'b.c': 1,  
          'd.tv': 1,  
          'spam.com?': 1}
```

```
In [20]: # call and test both the methods from the main method  
def main():  
    extractAllHostnames('emails.txt')  
    extractHostname('emails.txt',0)  
    extractAllDomains('emails.txt')  
    extractDomains('emails.txt',2)  
    extractHostname('big-emails.txt',1)  
main()
```

after you write the above 2 methods - **EXTEND** the functionality of this project by for example, adding more criterion for email/hostname check and think of more ways we can make this project more useful to users - So that they buy our App!

add at least 2 more methods of your own.