```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 19 14:35:38 2017

@author: Harsh Kevadia
"""

from bs4 import BeautifulSoup
import re
import requests
import random

def run(url):

    fin=open('tags.txt','r') # input file
    fw=open('questions.txt','w') # output file
    log=open('log.txt','w') # log file
    test=open('test.txt','w') # test file
    train = open('train.txt','w') # training file

    c = 0 #just a counter for questions scraped
    c1 = 0
    c2 = 0

    vote_cutoff = -1 #vote threshold
    pageNum=42 # number of pages to collect
    test_percent = 0.1 #

    for line in fin: #for each tag
        html=None
        tag=line.lower().strip()
        print(tag)
        log.write('tag: ')
        log.write(tag+'\n')

        t =0
        t1 = 0
        t2 = 0

        for n in range(pageNum):
            if n == 0: continue
            pageLink=url+tag+'-interview-questions&n='+str(n) # make the page url
            arr1 = []

            for i in range(5): # try 5 times
                try:
                    #use the browser to access the url
                    response=requests.get(pageLink,headers = { 'User-Agent': 'Mozilla/5.0 (Windows
                    html=response.content # get the html
                    break # we got the file, break the loop
                except Exception as e:# threw an exception, the attempt to get the response failed
                    print ('failed attempt',i)
                    #time.sleep(2) # wait 2 secs


            if not html:continue # couldnt get the page, ignore
            soup = BeautifulSoup(html.decode('ascii', 'ignore'),'lxml') # parse the html

            questions=soup.findAll('li', {'class':re.compile('question')}) # get all the question

            for question in questions:
                if question == questions[0]:
                    print('Page '+str(n))
                    log.write('Page '+str(n)+'\n')
                votes,text='NA','NA' # initialize votes and text
                votes = int(question.find('div', {'class':re.compile("votesNetQuestion")}).text)
                if votes > vote_cutoff :
                    text = question.find('p').text.replace("\r", " ").replace("\n", " ").replace("\
                    arr1.append(tag + "\t" + str(votes) + "\t" + text +'\n')
                    fw.write(tag + "\t" + str(votes) + "\t" + text +'\n')
                    c=c + 1
                    t = t + 1

            for i in range(int(len(arr1) * test_percent)):
                index = random.sample(range(len(arr1)),1)[0]
                test.write(arr1[index])
                arr1.remove(arr1[index])
                c1 = c1 + 1
                t1 = t1 + 1

            for i in range(len(arr1)):
                train.write(arr1[i])
                c2 = c2 + 1
                t2 = t2 + 1

        print('Questions Added to Tag:'+str(t))
        log.write('Questions Added to Tag:'+str(t)+'\n')
        print('Questions Added to Test:'+str(t1))
        log.write('Questions Added to Test:'+str(t1)+'\n')
        print('Questions Added to Train:'+str(t2))
        log.write('Questions Added to Train:'+str(t2)+'\n')
        print('Total Questions Added to Testing:'+str(c1))
        log.write('Total Questions Added to Testing:'+str(c1)+'\n')
        print('Total Questions Added to Training:'+str(c2))
        log.write('Total Questions Added to Training:'+str(c2)+'\n\n')

        print('Total Questions Scraped:'+str(c))
        log.write('Total Questions Scraped:'+str(c)+'\n')

    fin.close()
    fw.close()
    log.close()
    train.close()
    test.close()

if __name__=='__main__':
    url='https://www.careercup.com/page?pid='
    run(url)
```