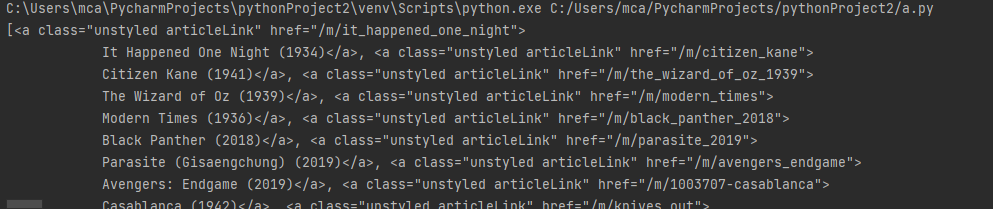
Program No:16

Date:16-02-2022

Aim: Implement a simple web crawler

import requests  
import lxml  
from bs4 import BeautifulSoup  
url="https://www.rottentomatoes.com/top/bestofrt/"  
headers={  
 'User-Agent':'Mozilla/5.0(windows NT 6.1;WOW64)AppleWebKit/537.36(KHTML,like Gecko)Chrome/63.0.3239.132 Safari/537.36 QITHU 360 SE'  
}  
f=requests.get(url,headers=headers)  
movies\_lst=[]  
soup =BeautifulSoup(f.content,'html.parser')  
movies=soup.find('table',{  
 'class':'table'  
}).find\_all('a')  
print(movies)  
num=0  
for anchor in movies:  
 urls="https://www.rottentomatoes.com"+anchor['href']  
 movies\_lst.append(urls)  
print(movies\_lst)  
num +=1  
movie\_url=urls  
movie\_f=requests.get(movie\_url,headers=headers)  
movie\_soup=BeautifulSoup(movie\_f.content,'lxml')  
movie\_content = movie\_soup.find('div',{  
 'class': 'movie\_synopsis clamp clamp-6 js-clamp'  
 })  
print(num,urls,'\n','movie:'+anchor.string.strip())  
print('Movie info:'+movie\_content.string.strip())

OUT PUT







Program No:17

Date:16-02-2022

Aim: Implement a simple web crawler(program two)

from bs4 import BeautifulSoup  
import requests  
pages\_creawled=[]  
def crawler(url):  
 page = requests.get(url)  
 soup = BeautifulSoup(page.text,'html.parser')  
 links=soup.find\_all('a')  
  
 for link in links:  
 if 'href' in link.attrs:  
 if link['href'].startswith('/WiKi') and ':' not in link['href']:  
 if link['href'] not in pages\_creawled:  
 new\_link = f"https://en.wikipedia.org{link['href']}"  
 pages\_creawled.append(link['href'])  
 try:  
 with open('data.csv,a') as file:  
 file.write(f'{soup.title.text};{soup.h1.text};{link["href"]}\n')  
 crawler(new\_link)  
 except:  
 continue  
  
  
crawler('https://en.wikipedia.org')

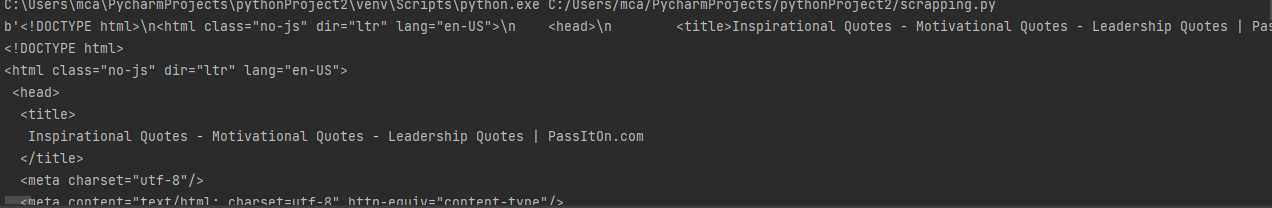
Program No:18

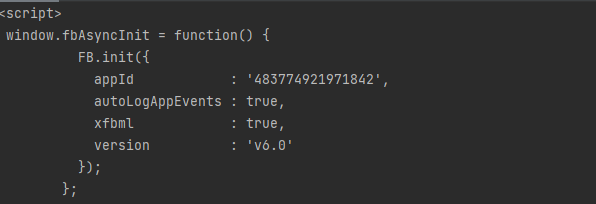
Date:16-02-2022

Aim: implement a program to to scrap the content of any web page

import requests  
from bs4 import BeautifulSoup  
import csv  
import lxml  
URL="http://www.values.com/inspirational-quotes"  
r = requests.get(URL)  
print(r.content)  
soup = BeautifulSoup(r.content,'lxml')  
print(soup.prettify())  
quotes=[]  
table = soup.find('div',attrs={'id':'all\_quotes'})  
for row in table.findAll('div',  
 attrs={'class':'col-6 col-log3 text-center margin-30px-bottom sm-margin-30px-top'}):  
 quote ={}  
 quote['theme']=row.h5.text  
 quote['url']=row.a['href']  
 quote['img']=row.img['src']  
 quote['lines']=row.img['alt'].split('#')[0]  
 quote['author']=row.img['alt'].split('#')[1]  
 quotes.append(quote)  
 filename='inspirational\_quotes.csv'  
 with open(filename,'w',newline='') as f:  
 W=csv.DictWriter(f,['theme','url','img','lines','author'])  
 W.writeheader()  
 for quote in quotes:  
 W.writerow(quote)

OUTPUT





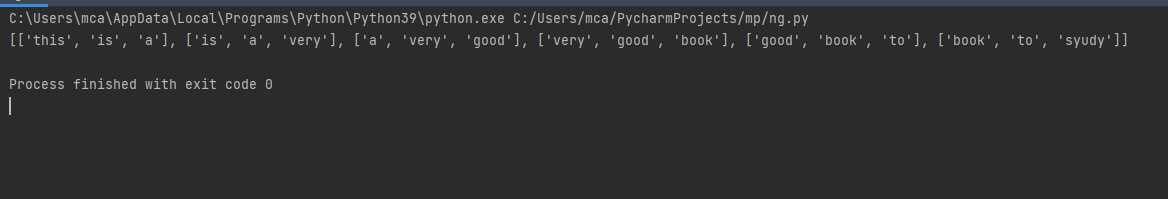
Program No:19

Date:16-02-2022

Aim: Python program for natural language processing using Ngram (without using inbuilt function)

Aim: ngram  
def generate\_ngrams(text,WordsToCombine):  
 words = text.split()  
 output= []  
 for i in range(len(words)- WordsToCombine + 1):  
 output.append(words[i:i + WordsToCombine])  
 return output  
x=generate\_ngrams(text='this is a very good book to syudy',WordsToCombine=3)  
print(x)

OUTPUT



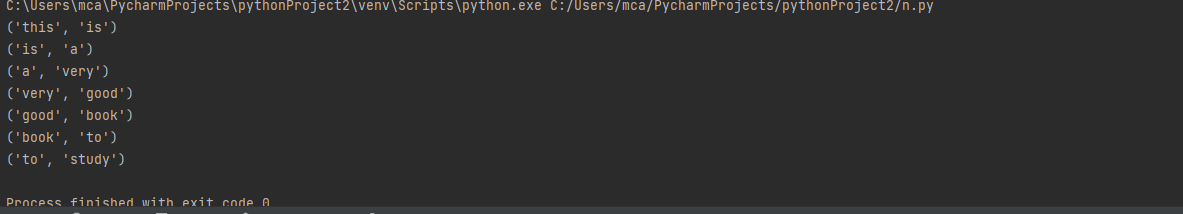
Program No:20

Date:16-02-2022

Aim: Python program for natural language processing using Ngram (with inbuilt function)

import nltk  
nltk.download()  
from nltk.util import ngrams  
samplText = 'this is a very good book to study'  
NGRAMS = ngrams(sequence=nltk.word\_tokenize(samplText),n=2)  
for grams in NGRAMS:  
 print(grams)

OUTPUT



Program No:21

Date:16-02-2022

Aim: Python program for natural language processing – speech tagging

import nltk  
from nltk.corpus import stopwords  
from nltk.tokenize import word\_tokenize,sent\_tokenize  
stop\_words = set(stopwords.words('english'))  
txt ="sukanya,ram are friends"\  
 "Cats are often kept as pets"\  
 "lions, tigers, and other wild animals in the same family."  
tokenize= sent\_tokenize(txt)  
for i in tokenize:  
 wordsList = nltk.word\_tokenize(i)  
 wordsList = [w for w in wordsList if not w in stop\_words]  
 tagged = nltk.pos\_tag(wordsList)  
 print(tagged)

OUTPUT

