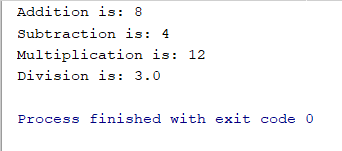
***ARTIFICAL INTELLIGENCE (LAB) MANUEL***

**LAB TASKS #1**

**Task-1:**

Write a python code to perform all four mathematical operations and display the result.

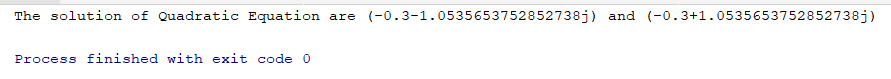
a = 6  
b = 2  
print(**"Addition is:"**, a+b)  
print(**"Subtraction is:"**, a-b)  
print(**"Multiplication is:"**, a\*b)  
print(**"Division is:"**, a/b)



**Task-2:**

Write down a python code to Solve the quadratic equation when a = 5, b = 3 and c = 6 and display the result.

**import** cmath  
a = 5  
b = 3  
c = 6  
d = (b\*\*2) - (4\*a\*c)  
sol1 = (-b-cmath.sqrt(d))/(2\*a)  
sol2 = (-b+cmath.sqrt(d))/(2\*a)  
print(**'The solution of Quadratic Equation are {0} and {1}'**.format(sol1,sol2))



**Task-3a:**

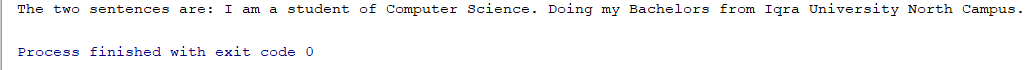
Write down a python code to use concatenation to join two different sentences together.

**Task-3b:**

Print the second word in the sentence generated in task 3a.

*Satisfying both 3a & 3b:*

sen1 = **"I am a student of Computer Science."**sen2 = **"Doing my Bachelors from Iqra University North Campus."**print(**"The two sentences are:"**, sen1, sen2)

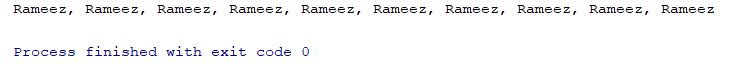
****

**LAB TASKS #2**

**Task 1a:**

Write down a python code to print your name 10 times in a single row.

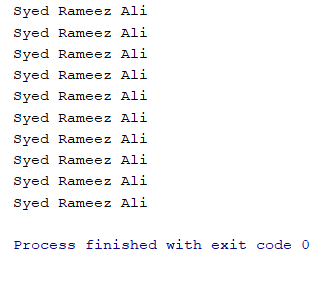
print(**', '**.join([str(**"Taha ali"**) **for** i **in** range(10)]))



**Task 1b:**

Write down a python code to print your name 10 times.

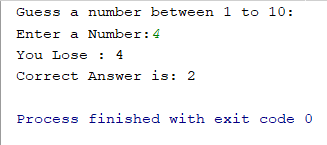
**for** i **in** range(0,10):  
 print(**"Hafiz Syed Taha Ali"**)



**Task 2:**

Write down a python code to create a guessing game in which computer will guess a number randomly and ask you to guess it correctly.

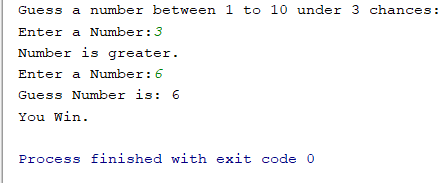
**import** random  
number = random.randint(1, 10)  
print(**"Guess a number between 1 to 10:"**)  
guess\_number = int(input(**"Enter a Number:"**))  
**if** guess\_number == number:  
 print(**"You Win :"**, guess\_number)  
**else**:  
 print(**"You Lose :"**, guess\_number)  
 print(**"Correct Answer is:"**, number)



**Task 3:**

Write down a python code to create a guessing game in which computer will guess a number randomly and ask you to guess it correctly with in three chance.

**import** random  
number = random.randint(1, 10)  
print(**"Guess a number between 1 to 10 under 3 chances:"**)  
**for** i **in** range(0, 4):  
 **if** i < 3:  
 guess\_number = int(input(**"Enter a Number:"**))  
 **if** guess\_number < number:  
 print(**"Number is greater."**)  
 **elif** guess\_number > number:  
 print(**"Number is lesser."**)  
 **elif** guess\_number == number:  
 print(**"Guess Number is:"**, guess\_number)  
 print(**"You Win."**)  
 **break  
 elif** i == 3:  
 print(**"Guess Number is:"**, number)  
 print(**"You Lose."**)

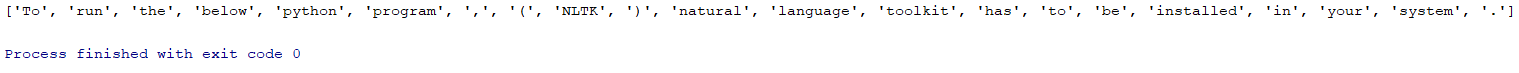


**LAB TASKS #3**

**Task-1a:**

Write down a python code to use nltk for breaking up a paragraph into words.

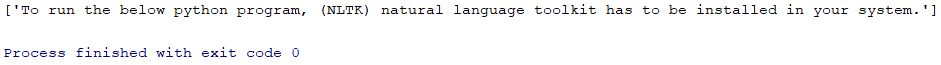
**import** nltk  
**from** nltk **import** word\_tokenize  
sample\_text = **"To run the below python program, (NLTK) natural language toolkit has to be installed in your system. The NLTK module is a massive tool kit, aimed at helping you with the entire Natural Language Processing (NLP) methodology."**print(word\_tokenize(sample\_text))



**Task-1b:**

Write down a python code to use nltk for breaking up a paragraph into sentences.

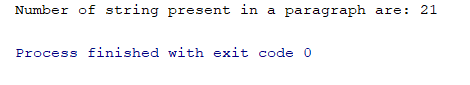
**import** nltk  
**from** nltk **import** sent\_tokenize  
sample\_text = **"To run the below python program, (NLTK) natural language toolkit has to be installed in your system."**print(sent\_tokenize(sample\_text))



**Task-2:**

Determine the number of the user define string present in the paragraph in task-1.

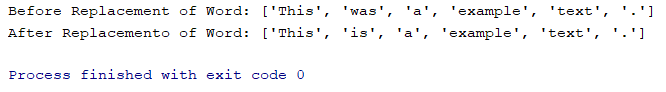
**import** nltk  
**from** nltk **import** word\_tokenize  
sample\_text = **"To run the below python program, (NLTK) natural language toolkit has to be installed in your system."**numberOf\_string = len(word\_tokenize(sample\_text))  
print(**"Number of string present in a paragraph are:"**, numberOf\_string)



**Task-3:**

Write down a python code to substitute a specific word of the whole sentences.

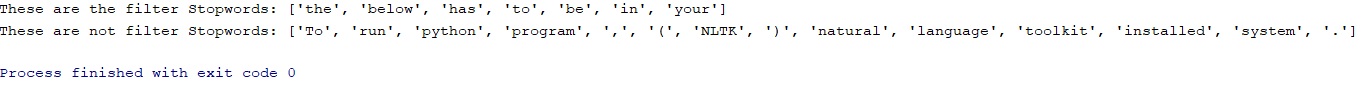
**import** nltk  
**from** nltk.tokenize **import** word\_tokenize  
sample\_text = **"This was a example text."**words = sample\_text.replace(**"was"**, **"is"**)  
print(**"Before Replacement of Word:"**, word\_tokenize(sample\_text))  
print(**"After Replacemento of Word:"**, word\_tokenize(words))



**Task-4:**

Write down a python code to filter the given sentence of stopwords.

**import** nltk  
**from** nltk.corpus **import** stopwords  
**from** nltk.tokenize **import** word\_tokenize  
  
sample\_text = **"To run the below python program,(NLTK) natural language toolkit has to be installed in your system."**stop\_words = set(stopwords.words(**"english"**))  
words = word\_tokenize(sample\_text)  
filter\_sentence1 = []  
filter\_sentence2 = []  
**for** w **in** words:  
 **if** w **in** stop\_words:  
 filter\_sentence1.append(w)  
 **elif** w **not in** stop\_words:  
 filter\_sentence2.append(w)  
print(**"These are the filter Stopwords:"**, filter\_sentence1)  
print(**"These are not filter Stopwords:"**, filter\_sentence2)

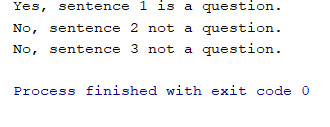


**LAB TASKS #4**

**Task-1:**

Write down a python code to identify the question in the given paragraph. ("Hello! Mr. Ali, How are you doing today? The weather is great and python is very good. The sky is blue and you should not eat then.")

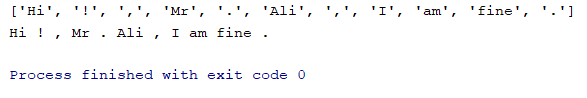
**import** nltk  
**from** nltk.tokenize **import** word\_tokenize, sent\_tokenize  
sample\_text = **"Hello!, Mr. Ali, How are you doing today? The weather is great and python is very good. The sky is blue and you should not eat then."**Answer = []  
sen = sent\_tokenize(sample\_text)  
sen1 = sen[0]  
sen2 = sen[1]  
sen3 = sen[2]  
sen1\_word = word\_tokenize(sen1)  
sen2\_word = word\_tokenize(sen2)  
sen3\_word = word\_tokenize(sen3)  
**if** sen1[-1] **in** sen1\_word:  
 print(**"Yes, sentence 1 is a question."**)  
  
**if "?" or "." in** sen2\_word:  
 print(**"No, sentence 2 not a question."**)  
  
**if "?" or "." in** sen3\_word:  
 print(**"No, sentence 3 not a question."**)



**Task-2:**

Write down a python code to identify the question and generate an appropriate response from task-1.

**import** nltk  
**from** nltk.tokenize **import** word\_tokenize, sent\_tokenize  
sample\_text = **"Hello!, Mr. Ali, How are you doing today? The weather is great and python is very good. The sky is blue and you should not eat then."**Answer = []  
sen = sent\_tokenize(sample\_text)  
sen1 = sen[0]  
sen2 = sen[1]  
sen3 = sen[2]  
sen1\_word = word\_tokenize(sen1)  
sen2\_word = word\_tokenize(sen2)  
sen3\_word = word\_tokenize(sen3)  
**if "?" in** sen1\_word:  
 **if** sen1\_word[0] **in** sen1\_word:  
 Answer.append(**"Hi"**)  
 **if** sen1\_word[1] **in** sen1\_word:  
 Answer.append(**"!"**)  
 **if** sen1\_word[2] **in** sen1\_word:  
 Answer.append(**","**)  
 **if** sen1\_word[3] **in** sen1\_word:  
 Answer.append(**"Mr"**)  
 **if** sen1\_word[4] **in** sen1:  
 Answer.append(**"."**)  
 **if** sen1\_word[5] **in** sen1\_word:  
 Answer.append(**"Ali"**)  
 **if** sen1\_word[6] **in** sen1\_word:  
 Answer.append(**","**)  
 **if** sen1\_word[7] **in** sen1\_word:  
 Answer.append(**"I"**)  
 **if** sen1\_word[8] **in** sen1\_word:  
 Answer.append(**"am"**)  
 **if** sen1\_word[9] **in** sen1\_word:  
 Answer.append(**"fine"**)  
 **if** sen1\_word[10] **in** sen1\_word:  
 Answer.append(**"."**)  
  
print(Answer)  
print(**" "**.join(Answer))

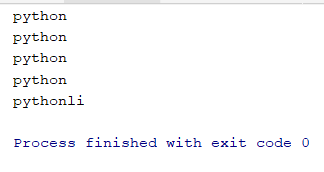


**LAB TASKS #5**

**Task-1:**

Write down a python code to perform stemming on the given words. (Take, Taking, Took, Taken).

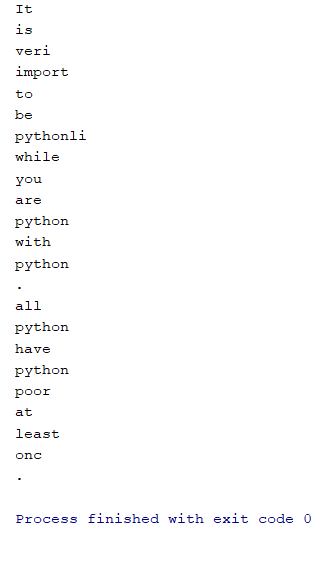
**import** nltk  
**from** nltk.stem **import** PorterStemmer  
  
pc = PorterStemmer()  
exp\_1 = [**"Python"**,**"Pythoner"**,**"Pythoning"**,**"Pythoned"**,**"Pythonly"**]  
**for** w **in** exp\_1:  
 print(pc.stem(w))



**Task-2:**

Write down a python code to perform stemming on the given sentence. (It is very important to be Pythonly while you are pythoning with python. All pythoners have pythoned poorly at least once.)

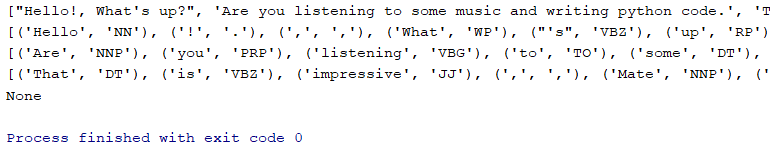
**import** nltk  
**from** nltk.stem **import** PorterStemmer  
**from** nltk **import** word\_tokenize  
  
pc = PorterStemmer()  
exp\_1 = [**"Python"**,**"Pythoner"**,**"Pythoning"**,**"Pythoned"**,**"Pythonly"**]  
  
sen\_1 = **"It is very important to be pythonly while you are pythoning with python. All pythoners have pythoned poor at least once."**words = word\_tokenize(sen\_1)  
  
**for** w **in** words:  
 print(pc.stem(w))

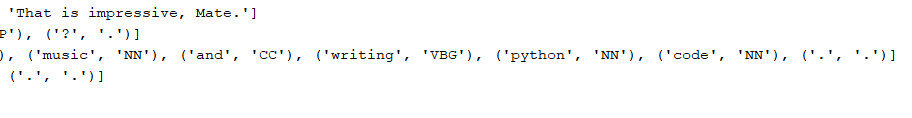


**Task-3:**

Write down python code to perform POS tagging on the given sentence.

**import** nltk  
**from** nltk.corpus **import** state\_union  
**from** nltk.tokenize **import** PunktSentenceTokenizer  
  
sample\_txt = **"Hello!, What's up? Are you listening to some music and writing python code. That is impressive, Mate."**custom\_sent\_tok = PunktSentenceTokenizer(sample\_txt) *#Pre-Train library the sample\_txt*print(custom\_sent\_tok)  
tokenized = custom\_sent\_tok.tokenize(sample\_txt)  
print(tokenized)  
  
**def** process\_cxt():  
 **try**:  
 **for** i **in** tokenized:  
 words = nltk.word\_tokenize(i)  
 tagg = nltk.pos\_tag(words)  
 print(tagg)  
 **except** Exception **as** e:  
 print(str(e))  
  
print(process\_cxt())



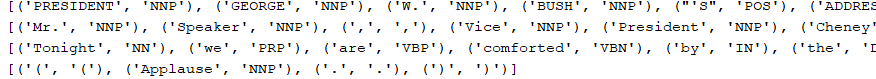


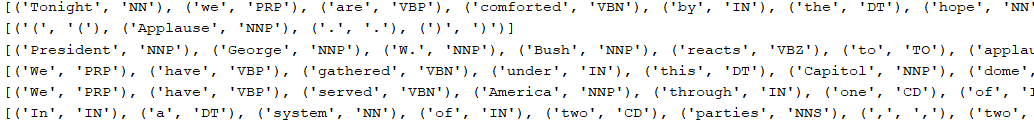
**Task-4:**

Write down python code to perform POS tagging on the state union speeches of gwbush-2006.

**import** nltk  
**from** nltk.corpus **import** state\_union  
**from** nltk.tokenize **import** PunktSentenceTokenizer  
  
train\_txt = state\_union.raw(**"2005-GWBush.txt"**)  
sample\_txt = state\_union.raw(**"2006-GWBush.txt"**)  
custom\_sent\_tok = PunktSentenceTokenizer(sample\_txt) *#Pre-Train library the sample\_txt*print(custom\_sent\_tok)  
tokenized = custom\_sent\_tok.tokenize(sample\_txt)  
print(tokenized)

**def** process\_cxt():  
 **try**:  
 **for** i **in** tokenized:  
 words = nltk.word\_tokenize(i)  
 tagg = nltk.pos\_tag(words)  
 print(tagg)  
 **except** Exception **as** e:  
 print(str(e))  
  
print(process\_cxt())





**LAB TASKS #6**

**Task-1A:**

Write down a Python code to determine the named entity in the text.

import nltk

from nltk.corpus import state\_union

from nltk.tokenize import PunktSentenceTokenizer

train\_text = state\_union.raw("2005-GWBush.txt")

sample\_text = state\_union.raw("2006-GWBush.txt")

custon\_sent\_tok = PunktSentenceTokenizer(train\_text)

tokenized = custon\_sent\_tok.tokenize(sample\_text)

def process\_content():

try:

for i in tokenized[5:]:

words = nltk.word\_tokenize((i))

tagged = nltk.pos\_tag(words)

namedEnt = nltk.ne\_chunk(tagged, binary=True)

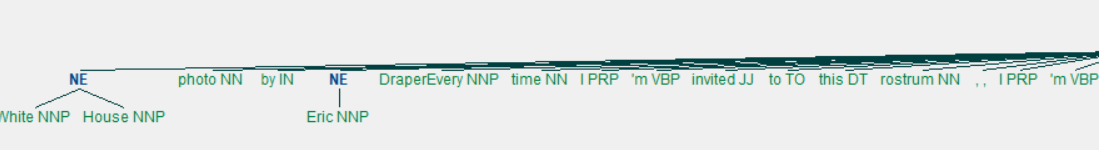
#namedEnt.draw()

print(namedEnt)

except Exception as s:

print(str(s))

process\_content()





**Task-1B:**

Write down a Python code to take a user define sentence and identify the named entity.

import nltk

from nltk.corpus import state\_union

from nltk.tokenize import PunktSentenceTokenizer

train\_text = state\_union.raw("2005-GWBush.txt")

sample\_text = "Hello!, My name is Ali Akhbar Siddiqui. I am here to teach a course of Artifical Intelligence in Iqra University."

custon\_sent\_tok = PunktSentenceTokenizer(train\_text)

tokenized = custon\_sent\_tok.tokenize(sample\_text)

def process\_content():

try:

for i in tokenized:

words = nltk.word\_tokenize((i))

tagged = nltk.pos\_tag(words)

namedEnt = nltk.ne\_chunk(tagged)

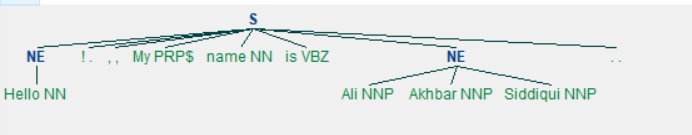
namedEnt.draw()

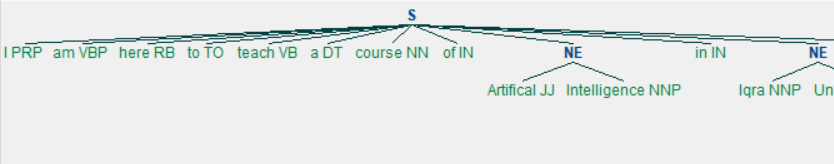
print(namedEnt)

except Exception as s:

print(str(s))

process\_content()

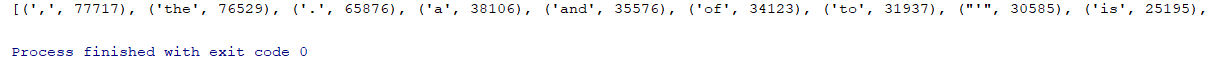




**Task-2A:**

Write down a python code to classify the text of the movie reviews from the corpus data set and identify the word appearing most of the time.

**import** nltk  
**import** random  
**from** nltk.corpus **import** movie\_reviews *# movie review data set*doc1 = [(list(movie\_reviews.words(fileid)), category)  
 **for** category **in** movie\_reviews.categories()  
 **for** fileid **in** movie\_reviews.fileids(category)]  
  
random.shuffle(doc1)  
  
all\_words = []  
**for** w **in** movie\_reviews.words():  
 all\_words.append(w.lower())  
  
all\_words = nltk.FreqDist(all\_words)  
print(all\_words.most\_common(30))



**Task-2B:**

Write down a python code to classify the text of the movie reviews from the corpus data set and identify the word appearing Least of the time.

**import** nltk  
**import** random  
**from** nltk.corpus **import** movie\_reviews *# movie review data set*doc1 = [(list(movie\_reviews.words(fileid)), category)  
 **for** category **in** movie\_reviews.categories()  
 **for** fileid **in** movie\_reviews.fileids(category)]  
  
random.shuffle(doc1)  
  
all\_words = []  
**for** w **in** movie\_reviews.words():  
 all\_words.append(w.lower())  
  
all\_words = nltk.FreqDist(all\_words)  
print(all\_words.most\_common(30) == False