Name: Harsh Zanwar Roll No:73 Branch:CSE(AIML)

Aim: Write a Python NLTK program to find the number of male and female names in the names corpus.Print the first 10 male and female names. Note: The names corpus contains a total of around 2943 male (male.txt) and 5001 female (female.txt) names. 2. Write a Python NLTK program to print the first 15 random combine labeled male and labeled female names from names corpus.

In [2]:
```
import nltk
import os
import nltk.corpus
```

In [3]:
```
print(os.listdir(nltk.data.find('corpora')))
```

```
['abc', 'abc.zip', 'alpino', 'alpino.zip', 'bcp47.zip', 'biocreative_ppi', 'b
iocreative_ppi.zip', 'brown', 'brown.zip', 'brown_tei', 'brown_tei.zip', 'ces
s_cat', 'cess_cat.zip', 'cess_esp', 'cess_esp.zip', 'chat80', 'chat80.zip',
'city_database', 'city_database.zip', 'cmudict', 'cmudict.zip', 'comparative_
sentences', 'comparative_sentences.zip', 'comtrans.zip', 'conll2000', 'conll2
000.zip', 'conll2002', 'conll2002.zip', 'conll2007.zip', 'crubadan', 'crubada
n.zip', 'dependency_treebank', 'dependency_treebank.zip', 'dolch', 'dolch.zi
p', 'europarl_raw', 'europarl_raw.zip', 'extended_omw.zip', 'floresta', 'flor
esta.zip', 'framenet_v15', 'framenet_v15.zip', 'framenet_v17', 'framenet_v17.
zip', 'gazetteers', 'gazetteers.zip', 'genesis', 'genesis.zip', 'gutenberg',
'gutenberg.zip', 'ieer', 'ieer.zip', 'inaugural', 'inaugural.zip', 'indian',
'indian.zip', 'jeita.zip', 'kimmo', 'kimmo.zip', 'knbc.zip', 'lin_thesaurus',
'lin_thesaurus.zip', 'machado.zip', 'mac_morpho', 'mac_morpho.zip', 'masc_tag
ged.zip', 'movie_reviews', 'movie_reviews.zip', 'mte_teip5', 'mte_teip5.zip',
'names', 'names.zip', 'nombank.1.0.zip', 'nonbreaking_prefixes', 'nonbreaking
_prefixes.zip', 'nps_chat', 'nps_chat.zip', 'omw-1.4.zip', 'omw.zip', 'opinio
n_lexicon', 'opinion_lexicon.zip', 'panlex_swadesh.zip', 'paradigms', 'paradi
gms.zip', 'pe08', 'pe08.zip', 'pil', 'pil.zip', 'pl196x', 'pl196x.zip', 'ppat
tach', 'ppattach.zip', 'problem_reports', 'problem_reports.zip', 'product_rev
iews_1', 'product_reviews_1.zip', 'product_reviews_2', 'product_reviews_2.zi
p', 'propbank.zip', 'pros_cons', 'pros_cons.zip', 'ptb', 'ptb.zip', 'qc', 'q
c.zip', 'reuters.zip', 'rte', 'rte.zip', 'semcor.zip', 'senseval', 'senseval.
zip', 'sentence_polarity', 'sentence_polarity.zip', 'sentiwordnet', 'sentiwor
dnet.zip', 'shakespeare', 'shakespeare.zip', 'sinica_treebank', 'sinica_treeb
ank.zip', 'smultron', 'smultron.zip', 'state_union', 'state_union.zip', 'stop
words', 'stopwords.zip', 'subjectivity', 'subjectivity.zip', 'swadesh', 'swad
esh.zip', 'switchboard', 'switchboard.zip', 'timit', 'timit.zip', 'toolbox',
'toolbox.zip', 'treebank', 'treebank.zip', 'twitter_samples', 'twitter_sample
s.zip', 'udhr', 'udhr.zip', 'udhr2', 'udhr2.zip', 'unicode_samples', 'unicode
_samples.zip', 'universal_treebanks_v20.zip', 'verbnet', 'verbnet.zip', 'verb
net3', 'verbnet3.zip', 'webtext', 'webtext.zip', 'wordnet.zip', 'wordnet2021.
zip', 'wordnet2022', 'wordnet2022.zip', 'wordnet31.zip', 'wordnet_ic', 'wordn
et_ic.zip', 'words', 'words.zip', 'ycoe', 'ycoe.zip']
```

```
In [4]: from nltk.corpus import brown
        brown.words()

Out[4]: ['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

```
In [5]: nltk.corpus.gutenberg.fileids()

Out[5]: ['austen-emma.txt',
         'austen-persuasion.txt',
         'austen-sense.txt',
         'bible-kjv.txt',
         'blake-poems.txt',
         'bryant-stories.txt',
         'burgess-busterbrown.txt',
         'carroll-alice.txt',
         'chesterton-ball.txt',
         'chesterton-brown.txt',
         'chesterton-thursday.txt',
         'edgeworth-parents.txt',
         'melville-moby_dick.txt',
         'milton-paradise.txt',
         'shakespeare-caesar.txt',
         'shakespeare-hamlet.txt',
         'shakespeare-macbeth.txt',
         'whitman-leaves.txt']
```

```
In [6]: emma = nltk.corpus.gutenberg.words('austen-emma.txt')
        len(emma)

Out[6]: 192427
```

```
In [21]: bible=nltk.corpus.gutenberg.words('bible-kjv.txt')
         bible

Out[21]: ['[', 'The', 'King', 'James', 'Bible', ']', 'The', ...]
```

```
In [8]: nltk.download('stopwords')

        [nltk_data] Downloading package stopwords to
        [nltk_data]     C:\Users\ASUS\AppData\Roaming\nltk_data...
        [nltk_data]   Package stopwords is already up-to-date!

Out[8]: True
```

```
In [9]: nltk.download('punkt')

        [nltk_data] Downloading package punkt to
        [nltk_data]     C:\Users\ASUS\AppData\Roaming\nltk_data...
        [nltk_data]   Package punkt is already up-to-date!

Out[9]: True
```

In [10]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

In [13]:
```
stopwords=set(stopwords.words('english'))
```

In [14]:
```
stopwords
```

Out[14]:
```
{'a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'ain',
 'all',
 'am',
 'an',
 'and',
 'any',
 'are',
 'aren',
 "aren't",
 'as',
 'at',
 'be',
 'because',
```

In [15]:
```
ex='Hello, How are you ? I am fine.'
```

In [27]:
```
tokenize=word_tokenize(ex)
```

In [28]:
```
tokenize
```

Out[28]:
```
['Hello', ',', 'How', 'are', 'you', '?', 'I', 'am', 'fine', '.']
```

In [19]:
```
wordfilter=[w for w in tokenize if not w in stopwords]
wordfilter
```

Out[19]:
```
['Hello', ',', 'How', '?', 'I', 'fine', '.']
```

In [23]:
```
filtered_sen=[]
for w in tokenize:
    if w not in stopwords:
        filtered_sen.append(w)

filtered_sen
```

Out[23]:
```
['Hello', ',', 'How', '?', 'I', 'fine', '.']
```

In [24]:
```
text="Hello, How are you ? I am fine."
tokenizer=nltk.RegexpTokenizer(r"\w+")
tokens=tokenizer.tokenize(text)
" ".join(tokens)
```

Out[24]:
```
'Hello How are you I am fine'
```

In [26]:
```
stopwords=list(stopwords)
my_extra=['I','Apple','google']
stopwords.extend(my_extra)
stopwords
```

Out[26]:
```
['haven',
 'shouldn',
 'himself',
 'same',
 'did',
 'themselves',
 'in',
 "needn't",
 'just',
 'don',
 'have',
 'until',
 'if',
 'now',
 'out',
 'how',
 'again',
 'are',
 'a',
```

In [29]:
```
ex='Hello , I am fine.'
tokenize=word_tokenize(ex)
ex
```

Out[29]:
```
'Hello , I am fine.'
```

In [30]:
```
wordfilter=[w for w in tokenize if not w in stopwords]
wordfilter
```

Out[30]:
```
['Hello', ',', 'fine', '.']
```

In [31]:
```
stopwords=[el for el in stopwords if el not in my_extra]
```

In [32]:
```
stopwords
```

Out[32]:
```
['haven',
 'shouldn',
 'himself',
 'same',
 'did',
 'themselves',
 'in',
 "needn't",
 'just',
 'don',
 'have',
 'until',
 'if',
 'now',
 'out',
 'how',
 'again',
 'are',
 'a',
```

#stemming -To reduce the root words.

In [33]:
```
from nltk.stem import PorterStemmer
```

In [34]:
```
from nltk.tokenize import sent_tokenize,word_tokenize
```

In [35]:
```
ps=PorterStemmer()
```

In [36]:
```
ex_word=["python","pythoner","pythoning","pythoned","pythons"]
```

In [38]:
```
for w in ex_word:
    print(ps.stem(w))
```

```
python
python
python
python
python
```

In [39]:
```
import nltk
from nltk import word_tokenize
text="This is one simple example."
tokens=word_tokenize(text)
tags=nltk.pos_tag(tokens,tagset="universal")
tags
```

Out[39]:
```
[('This', 'DET'),
 ('is', 'VERB'),
 ('one', 'NUM'),
 ('simple', 'ADJ'),
 ('example', 'NOUN'),
 ('.', '.')]
```

In [ ]:

In [ ]: