NAME :- MOHAN SAI KOTHAPALLI **EMP ID :- 2112951** COHORT CODE: - GN22CDBDS001 ASSIGNMENT ON REGEX Design a python program to accept a file name through command line arguments. Parse this file to perform the following: Print all currencies in text, Accepted-\$, ₹, £ 2. Print all date times in the text- dd/mm/yyyy, dd/mm/yy, mm/dd/yyyy, mm/dd/yy 3. Print all cardinilities and orders- 4th, fifth, sixth, 1st, 2nd, nineteenth, fifth 4. Print all 4 letter words that begin with vowels STEP 0:- IMPORTING LIBRARIES In [1]: import re import sys DATA 1 STEP 1:- Opening File DATA 1 This file contains some random text with currency's and dates and some orders. In [2]: f=open('data1.txt') In [3]: with open('data1.txt','r',encoding='utf-8') as f: data=f.readline() data Out[3]: 'Indian currency symbol is ₹. USA currency symbol is \$. UK currency symbol is £. 1₹ is 0.013\$ and 1\$ is 75.95₹ as of 02/04/2022. 1₹ is 0.010£ and 1£ is 99.61£ as of 04/02/2022. 1\$ is 76.048₹ as of 28/03/22. 1\$ is 75.998₹ as of 03/20/22. ISO 8601 allows for date representations without the hyphens sepa rating the parts of the date. For example 14th June 2021 can be written as 2021-06-14 in the extended human-readable format but also as 20210614 in the basic format. If agreed upon by both the sender and receiver, we can modify the year representation to include additional year characters. For example, i f we'd like to include a 5th year-character and fourteenth hundreeth fifth ours Ours first 1st.\n' The above 'readline()' will read a single line from the file. Line means when you press "enter" in the keyboard will typing your data that is considered as new line. Without pressing you keep on typing then it is considered as single line. i.e '\n' with open('data1.txt','r',encoding='utf-8') as f: In [4]: data=f.readlines() data[:5] Out[4]: ['Indian currency symbol is ₹. USA currency symbol is \$. UK currency symbol is £. 1₹ is 0.013\$ and 1\$ is 75.95₹ as of 02/04/2022. 1₹ is 0.010£ and 1£ is 99.61£ as of 04/02/2022. 1\$ is 76.048₹ as of 28/03/22. 1\$ is 75.998₹ as of 03/20/22. ISO 8601 allows for date representations without the hyphens sepa rating the parts of the date. For example 14th June 2021 can be written as 2021-06-14 in the extended human-readable format but also as 20210614 in the basic format. If agreed upon by both the sender and receiver, we can modify the year representation to include additional year characters. For example, i f we'd like to include a 5th year-character and fourteenth hundreeth fifth ours Ours first 1st.\n', 'In addition, this does not rule out invalid leap days. For example, it will match 02/29/2021 althou gh 2021 is not a leap year. To do this check will require checking if the year is divisible by 4. Thi s is not possible with regular expressions and it is recommended that you use the date tools in your programming language of choice to check for validity.\n', 'The column Independent Filing Deadline shows the date for the filing of petitions by independent or third/minor party candidates. This is a general reference\n', 'date for use by the public and voters. Candidates and others seeking specific information should co ntact the states for other deadlines that may need $\n'$ , 'to be met. For example, the petitions may have to be checked by officials prior to this date. A dec laration of candidacy may be due before the petitions are\n'] The above 'readlines()' will read all lines from the file as list. Each line as an element in list. with open('data1.txt','r',encoding='utf-8') as f: In [5]: data[:1000] Out[5]: 'Indian currency symbol is ₹. USA currency symbol is \$. UK currency symbol is £. 1₹ is 0.013\$ and 1\$ is 75.95₹ as of 02/04/2022. 1₹ is 0.010£ and 1£ is 99.61£ as of 04/02/2022. 1\$ is 76.048₹ as of 28/03/22. 1\$ is 75.998₹ as of 03/20/22. ISO 8601 allows for date representations without the hyphens sepa rating the parts of the date. For example 14th June 2021 can be written as 2021-06-14 in the extended human-readable format but also as 20210614 in the basic format. If agreed upon by both the sender and receiver, we can modify the year representation to include additional year characters. For example, i f we'd like to include a 5th year-character and fourteenth hundreeth fifth ours Ours first 1st.\nIn a ddition, this does not rule out invalid leap days. For example, it will match 02/29/2021 although 202 1 is not a leap year. To do this check will require checking if the year is divisible by 4. This is n ot possible with regular expressions and it is recommended that you use the date tools in y' The above 'read()' will read all lines from the file as single para. STEP 2:- FINDING CURRENCY'S INCLUDING THE AMOUNT IN TEXT In [6]:  $x=re.findall(r''(\d^*?\.?\d^+?[\$?])'', data)$ In [7]: print(x) ['1₹', '0.013\$', '1\$', '75.95₹', '1₹', '0.010£', '1£', '99.61£', '1\$', '76.048₹', '1\$', '75.998₹'] FINDING ONLY THE SYMBOLS OF CURRENCY IN TEXT In [8]: curr=re.findall("([\$₹£])",data) In [9]: print("Total Number Of Currency Symbols In the TEXT DATA are : ",len(curr)) print(f"Types Of Currency Symbols In the TEXT DATA are : ",len(set(curr))," ",set(curr)) print(curr) Total Number Of Currency Symbols In the TEXT DATA are : 15Types Of Currency Symbols In the TEXT DATA are : 3 {'\$', '£', '₹'} STEP 3 :-PRINTING ALL THE FORMATES OF DATES IN THE TEXT In [10]: dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d{4})\b)", data) print("The Number of dates in the format of 'mm/dd/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' print()  $dates = re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d\{4\})\b)", data)$ print("The Number of dates in the format of 'dd/mm/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d\{2\})\b)", data)$ print("The Number of dates in the format of 'dd/mm/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d\{2\})\b)",data)$ print("The Number of dates in the format of 'mm/dd/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print() The Number of dates in the format of 'mm/dd/yyyy' are: 12 02/04/2022 04/02/2022 02/29/2021 04/23/2022 03/06/2022 11/08/2022 12/10/2022 07/22/2022 04/2 3/2022 04/23/2022 03/12/2022 03/12/2022 The Number of dates in the format of 'dd/mm/yyyy' are: 7 02/04/2022 04/02/2022 03/06/2022 11/08/2022 12/10/2022 03/12/2022 03/12/2022 The Number of dates in the format of 'dd/mm/yy' are : 3 28/03/22 03/05/20 03/08/22 The Number of dates in the format of 'mm/dd/yy' are : 4 03/20/22 03/05/20 03/19/22 03/08/22 STEP 4:- PRINTING ALL CARDINILITIES AND ORDERS FROM THE TEXT In [11]: | order=[]  $x=re.findall(r"((first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))|(fif[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|rd|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|rd|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))| \\ (fo$ h|st|nd|rd) | (fi[a-z]+th) | (six[a-z]+(th|st|nd|rd)) | (sev[a-z]+(th|st|nd|rd)) | (eig[a-z]+(th|st|nd|rd)) | (n = 1 ine[a-z] + (th|st|nd|rd)) + (ten[a-z] + (th|st|nd|rd)) + (ele[a-z] + (th|st|nd|rd)) + (th|st|nd|rd)) + (th|st|nd|rd)st|rd)))",data) for i in range(len(x)): if(x[i][0] not in order): order.append(x[i][0]) #print(order) orders=[] x1=re.findall(r"([0-9]+(th|st|nd|rd))",data)for i in range(len(x1)): if(x1[i][0] not in orders):orders.append(x1[i][0]) #print(orders) if order or orders: print(order+orders) else: print("No CARDINILITIES AND ORDERS FOUND IN THE TEXT") ['fourteenth', 'hundreeth', 'fifth', 'first', 'third', '14th', '5th', '1st'] STEP 5 :- PRINTING ALL 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT In [12]:  $four=re.findall(r"(\b(a|e|i|o|u|A|E|I|O|U)[a-zA-Z]{3}\b)", data)$ for i in range(len(four)): fours.append(four[i][0]) print(fours) ['also', 'upon', 'ours', 'Ours', 'Utah', 'Utah'] WE WILL REPEAT THE ABOVE STEPS FOR THE REMAINING DATASETS DATA 2 STEP 1 :- Opening File DATA 2 This TEXT file contains a story and some values of currency's. In [13]: f=open('data2.txt') In [14]: | with open('data2.txt','r',encoding='utf-8') as f: data=f.readline() data Out[14]: "'Well, Mr Smith, first if you prefer a different type of nose, we have a large selection availabl e.'\n" In [15]: with open('data2.txt','r',encoding='utf-8') as f: data=f.readlines() data[:10] Out[15]: ["'Well, Mr Smith, first if you prefer a different type of nose, we have a large selection availabl '\n', "'I think this nose is a bit too small.'\n", "'Small noses are very fashionable this year, Mr Smith, very fashionable.'\n", "'Do you think it suits me?' asked Mr Smith.\n", "'I think it second and third looks very nice,' said the shop assistant.\n", '\n'] In [16]: with open('data2.txt','r',encoding='utf-8') as f: data=f.read() data[:1000] Out[16]: "'Well, Mr Smith, first if you prefer a different type of nose, we have a large selection availabl e.'\n\n'I think this nose is a bit too small.'\n\n'Small noses are very fashionable this year, Mr Smi th, very fashionable.' $\n\$  you think it suits me?' asked Mr Smith. $\n\$  think it second and third looks very nice,' said the shop assistant. $\n\n'OK$ , I'll take it!' $\n\nOn$  the airbus home, Mr Smith cal led his wife on his wristphone.\n\n'Hello dear! Do you like my new nose?'\n\nMrs Smith looked at her husband's new nose on the videophone monitor on the wall in the kitchen. 'I think it's a bit too smal 1, dear,' she said.\n\n'Small noses are very fashionable this year,' replied Mr Smith, 'very fashiona ble.' It's all so easy now, thought Mr Smith. A hundred years ago, it was impossible to change your b ody. Or almost impossible - there was the old-fashioned 'plastic surgery', but it was expensive, pain ful and dangerous. Ugh! Now, thanks to our 22nd-century genetic engineering, we can change our bodies when we wa" STEP 2:- FINDING CURRENCY'S INCLUDING THE AMOUNT IN TEXT In [17]:  $x=re.findall(r''(\d^*?\.?\d^+?[\$?])'', data)$ In [18]: Out[18]: ['100000\$', '100000\$'] FINDING ONLY THE SYMBOLS OF CURRENCY IN TEXT In [19]: curr=re.findall("([\$₹£])", data) In [20]: print("Total Number Of Currency Symbols In the TEXT DATA are: ",len(curr)) print(f"Types Of Currency Symbols In the TEXT DATA are : ",len(set(curr))," ",set(curr)) print(curr) Total Number Of Currency Symbols In the TEXT DATA are: 2 Types Of Currency Symbols In the TEXT DATA are : 1 {'\$'} ['\$', '\$'] STEP 3:-PRINTING ALL THE FORMATES OF DATES IN THE TEXT In [21]:  $dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(d{4}))b)", data)$ print("The Number of dates in the format of 'mm/dd/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d{4})\b)",data)$ print("The Number of dates in the format of 'dd/mm/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d{2})\b)",data)$ print("The Number of dates in the format of 'dd/mm/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d\{2\})\b)", data)$ print("The Number of dates in the format of 'mm/dd/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print() The Number of dates in the format of 'mm/dd/yyyy' are: 0 The Number of dates in the format of 'dd/mm/yyyy' are : 0 The Number of dates in the format of 'dd/mm/yy' are: 0 The Number of dates in the format of 'mm/dd/yy' are : 0 STEP 4 :- PRINTING ALL CARDINILITIES AND ORDERS FROM THE TEXT In [22]: order=[] x=re.findall(r"((first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))|(fif[a-z]+(th|rd|st|nd))|h|st|nd|rd) | (fi[a-z]+th) | (six[a-z]+(th|st|nd|rd)) | (sev[a-z]+(th|st|nd|rd)) | (eig[a-z]+(th|st|nd|rd)) | (n = 1) | (n ine[a-z] + (th|st|nd|rd)) + (ten[a-z] + (th|st|nd|rd)) + (ele[a-z] + (th|st|nd|rd)) + (ten[a-z] +st|rd)))",data) for i in range(len(x)): if(x[i][0] not in order):order.append(x[i][0]) #print(order) orders=[] x1=re.findall(r"([0-9]+(th|st|nd|rd))",data)for i in range(len(x1)): **if**(x1[i][0] **not in** orders): orders.append(x1[i][0]) #print(orders) if order or orders: print(order+orders) print("No CARDINILITIES AND ORDERS FOUND IN THE TEXT") ['first', 'second', 'third', '22nd'] STEP 5 :- PRINTING ALL 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT In [23]:  $|four=re.findall(r"(\b(a|e|i|o|u|A|E|I|O|U)[a-zA-Z]{3}\b)", data)$ fours=[] for i in range(len(four)): fours.append(four[i][0]) ['easy', 'only', 'else', 'ears', 'Ears', 'ears', 'ears', 'ears', 'eyes', 'arms', 'ears', 'eyes', 'arm s', 'only', 'used', 'used', 'eyes', 'into'] DATA 3 STEP 1 :- Opening File DATA 3 This TEXT file contains INDIAN CONSTUITIONAL AMENDAMENTS with dates and order of the AMENDEMETS. As there are no currencys in this file i have included some random currency symbols with and without amounts (4) In [24]: f=open('data3.txt') In [25]: with open('data3.txt','r',encoding='utf-8') as f: data=f.readline() data Out[25]: '1st\t15, 19, 85, 87, 174, 176, 341, 342, 372 and 376. Insert articles 31A and 31B. Insert schedule 9.\t18/06/1951 \tAdded special provision for the advancement of any socially and educationally [7] ba ckward classes or for the Scheduled Castes and Scheduled Tribes (SCs and STs). To fully secure the co nstitutional validity of zamindari abolition laws and to place reasonable restriction on freedom of s peech. A new constitutional device, called Schedule 9 introduced to protect against laws that are con trary to the Constitutionally guaranteed fundamental rights. These laws encroach upon property right s, freedom of speech and equality before law.\tJawaharlal Nehru\tRajendra Prasad\n' In [26]: with open('data3.txt','r',encoding='utf-8') as f: data=f.readlines() data[:5] Out[26]: ['1st\t15, 19, 85, 87, 174, 176, 341, 342, 372 and 376. Insert articles 31A and 31B. Insert schedule 9.\t18/06/1951 \tAdded special provision for the advancement of any socially and educationally [7] ba ckward classes or for the Scheduled Castes and Scheduled Tribes (SCs and STs). To fully secure the co nstitutional validity of zamindari abolition laws and to place reasonable restriction on freedom of s peech. A new constitutional device, called Schedule 9 introduced to protect against laws that are con trary to the Constitutionally guaranteed fundamental rights. These laws encroach upon property right s, freedom of speech and equality before law.\tJawaharlal Nehru\tRajendra Prasad\n',  $'2nd\tAmend$  article 81(1)(b).[8]\t01/05/1953\tRemoved the upper population limit for a parliamentary constituency by amending Article 81(1)(b).\n', '3rd\tAmend schedule 7.[9]\t22/02/1955\tRe-enacted entry 33 of the Concurrent List in the Seventh Sc hedule with relation to include trade and commerce in, and the production, supply and distribution of four classes of essential commodities, viz., foodstuffs, including edible oil seeds and oils; cattle fodder, including oilcakes and other concentrates; raw cotton whether ginned or unginned, and cotton seeds; and raw jute. 100\$\n', '4th\tAmend articles £ 31, 35 and 305.\n', 'Amend schedule 9.[10]\t27/04/1955\tRestrictions on property rights and inclusion of related bills i n Schedule 9 of the constitution.\n'] In [27]: with open('data3.txt','r',encoding='utf-8') as f: data=f.read() data[:1000] Out[27]: '1st\t15, 19, 85, 87, 174, 176, 341, 342, 372 and 376. Insert articles 31A and 31B. Insert schedule 9.\t18/06/1951 \tAdded special provision for the advancement of any socially and educationally [7] ba ckward classes or for the Scheduled Castes and Scheduled Tribes (SCs and STs). To fully secure the co nstitutional validity of zamindari abolition laws and to place reasonable restriction on freedom of s peech. A new constitutional device, called Schedule 9 introduced to protect against laws that are con trary to the Constitutionally guaranteed fundamental rights. These laws encroach upon property right s, freedom of speech and equality before law.\tJawaharlal Nehru\tRajendra Prasad\n2nd\tAmend article 81(1)(b).[8]\t01/05/1953\tRemoved the upper population limit for a parliamentary constituency by amen ding Article 81(1)(b). $\n3rd\tAmend$  schedule 7.[9] $\t22/02/1955\tRe-enacted$  entry 33 of the Concurrent List in the Seventh Schedule with relation to include trade and commerce in, and the production, supp ly and d' STEP 2:- FINDING CURRENCY'S INCLUDING THE AMOUNT IN TEXT In [28]:  $x=re.findall(r"(\d^*?\.?\d+?[$₹£])", data)$ In [29]: x Out[29]: ['100\$', '100₹'] FINDING ONLY THE SYMBOLS OF CURRENCY IN TEXT In [30]: curr=re.findall("([\$₹£])",data) In [31]: print("Total Number Of Currency Symbols In the TEXT DATA are: ",len(curr)) print(f"Types Of Currency Symbols In the TEXT DATA are : ",len(set(curr))," ",set(curr)) print(curr) Total Number Of Currency Symbols In the TEXT DATA are : 4Types Of Currency Symbols In the TEXT DATA are : 3 {'\$', '£', '₹'} ['\$', '£', '₹', '\$'] STEP 3 :-PRINTING ALL THE FORMATES OF DATES IN THE TEXT In [32]: dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d{4})\b)", data) print("The Number of dates in the format of 'mm/dd/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d\{4\})\b)", data)$ print("The Number of dates in the format of 'dd/mm/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d\{2\})\b)", data)$ print("The Number of dates in the format of 'dd/mm/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d\{2\})\b)",data)$ print("The Number of dates in the format of 'mm/dd/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print() The Number of dates in the format of 'mm/dd/yyyy' are : 26 01/05/1953 12/24/1955 11/09/1956 01/11/1956 05/01/1960 12/28/1960 11/08/1961 01/12/1962 05/1 0/1963 05/10/1963 11/12/1966 10/04/1967 05/11/1971 08/12/1971 09/06/1972 09/06/1972 10/17/197  $3 \quad 01/06/1974 \quad 07/09/1974 \quad 01/03/1975 \quad 04/26/1975 \quad 03/05/1975 \quad 01/08/1975 \quad 07/08/1976 \quad 01/04/1977 \quad 01/08/1977 \quad 01/08/1979 \quad$ 6/09/1979 The Number of dates in the format of 'dd/mm/yyyy' are : 40 18/06/1951 01/05/1953 22/02/1955 27/04/1955 11/09/1956 01/11/1956 05/01/1960 11/08/1961 19/1 2/1961 20/12/1961 01/12/1962 28/12/1962 05/10/1963 05/10/1963 20/06/1964 27/08/1966 11/12/196 6 22/12/1966 10/04/1967 25/09/1969 23/12/1970 05/11/1971 08/12/1971 28/12/1971 30/12/1971 1 5/02/1972 29/08/1972 09/06/1972 09/06/1972 01/06/1974 19/05/1974 07/09/1974 01/03/1975 03/05/ 1975 01/08/1975 07/08/1976 01/04/1977 13/04/1978 06/09/1979 25/01/1980 The Number of dates in the format of 'dd/mm/yy' are : 2 10/08/75 27/05/76 The Number of dates in the format of 'mm/dd/yy' are: 1 10/08/75 STEP 4:- PRINTING ALL CARDINILITIES AND ORDERS FROM THE TEXT In [33]: order=[] x=re.findall(r"((first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))|(fif[a-z]+(th|rd|st|nd))|h|st|nd|rd) | (fi[a-z]+th) | (six[a-z]+(th|st|nd|rd)) | (sev[a-z]+(th|st|nd|rd)) | (eig[a-z]+(th|st|nd|rd)) | ine[a-z]+(th|st|nd|rd))|(ten[a-z]+?(th|st|nd|rd))|(ele[a-z]+th)|(twe[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[a-z]+(th|st|nd|rd))|(hun[st|rd)))",data) for i in range(len(x)): **if**(x[i][0] **not in** order): order.append(x[i][0]) #print(order) orders=[] x1=re.findall(r"([0-9]+(th|st|nd|rd))",data)for i in range(len(x1)): **if**(x1[i][0] **not in** orders): orders.append(x1[i][0]) #print(orders) if order or orders: print(order+orders) else: print ("No CARDINILITIES AND ORDERS FOUND IN THE TEXT") ['first', '1st', '2nd', '3rd', '4th', '5th', '6th', '7th', '8th', '9th', '10th', '11th', '12th', '13t h', '14th', '15th', '16th', '17th', '18th', '19th', '20th', '21st', '22nd', '23rd', '24th', '25th', '26th', '27th', '28th', '29th', '30th', '31st', '32nd', '33rd', '34th', '35th', '36th', '37th', '38t h', '39th', '40th', '41st', '42nd', '43rd', '44th', '45th'] STEP 5 :- PRINTING ALL 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT In [34]: | four=re.findall(r"(\b(a|e|i|o|u|A|E|I|O|U)[a-zA-Z]{3}\b)", data) fours=[] for i in range(len(four)): fours.append(four[i][0]) print(fours) print("Tolal number of words without repetition are : ",len(set(fours)),"\n",set(fours)) ['upon', 'oils', 'Also', 'into', 'over', 'away', 'into', 'into', 'acts', 'East', 'acts', 'into', 'act s', 'Anti'] Tolal number of words without repetition are: 9 {'upon', 'into', 'over', 'away', 'Anti', 'Also', 'acts', 'East', 'oils'} DATA 4 STEP 1 :- Opening File DATA 4 This TEXT file contains fluctuation of indian currency with dollar and pound daywise. In [35]: f=open('data4.txt') In [36]: with open('data4.txt','r',encoding='utf-8') as f: data=f.readline() data Out[36]: 'Date\t US Dollar to Indian Rupee\tLink\n' In [37]: with open('data4.txt','r',encoding='utf-8') as f: data=f.readlines() data[:10] Out[37]: ['Date\t US Dollar to Indian Rupee\tLink\n', 'Saturday 2nd April 2022\t1 \$ = 75.957 ₹\t\$ ₹ rate for 02/04/2022\n', 'Friday 1st April 2022\t 1 \$ = 75.953  $\neq \t$ \$  $\neq$  rate for  $01/04/2022\n'$ , 'Thursday 31st March 2022\t1  $$ = 75.910 ₹ t$ ₹ rate for 31/03/2022\n',$ 'Wednesday 30th March 2022\t1 \$ = 75.825 ₹\t\$ ₹ rate for 30/03/2022\n', 'Tuesday 29th March 2022\t1 \$ = 75.667 ₹ rate for 29/03/2022\n', 'Monday 28th March 2022\t1  $$ = 76.048 ₹ t$ ₹ rate for 28/03/2022\n',$ 'Sunday 27th March 2022\t1  $\$ = 76.287 ₹ t$ ₹ rate for 27/03/2022\n',$ 'Saturday 26th March 2022\t1 \$ = 76.275 ₹\t\$ ₹ rate for  $26/03/2022\n'$ , 'Friday 25th March 2022\t1 \$ = 76.276  $\gtrless$ \t\$  $\gtrless$  rate for 25/03/2022\n'] In [38]: with open('data4.txt','r',encoding='utf-8') as f: data=f.read() data[:1000] Out[38]: 'Date\t US Dollar to Indian Rupee\tLink\nSaturday 2nd April 2022\t1 \$ = 75.957 ₹\t\$ ₹ rate for 02/04/2022\nFriday 1st April 2022\t 1 \$ = 75.953 ₹\t\$ ₹ rate for 01/04/2022\nThursday 31st March 2022\t1 \$ = 75.910 ₹\t\$ ₹ rate for 31/03/2022\nWednesday 30th March 2 022\t1 \$ = 75.825 ₹\t\$ ₹ rate for 30/03/2022\nTuesday 29th March 2022\t1 \$ = 75.667 ₹\t\$ ₹ rate for 2 9/03/2022\nMonday 28th March 2022\t1 \$ = 76.048 ₹\t\$ ₹ rate for 28/03/2022\nSunday 27th March 2022\t1  $\$ = 76.287 \ \text{?}\ \text{t}\ \text{?}\ \text{rate for } 27/03/2022\nSaturday 26th March 2022\t1}\ \$ = 76.275 \ \text{?}\ \text{t}\ \text{?}\ \text{rate for } 26/03/2022\nSaturday 26th March 2022\t1}\$ 022\nFriday 25th March 2022\t1 \$ = 76.276 ₹\t\$ ₹ rate for 25/03/2022\nThursday 24th March 2022\t1 \$ = 76.318 ₹\t\$ ₹ rate for 24/03/2022\nWednesday 23rd March 2022\t1 \$ = 76.545 ₹\t\$ ₹ rate for 23/03/2022 \nTuesday 22nd March 2022\t1 \$ = 76.112 ₹\t\$ ₹ rate for 03/22/2022\nMonday 21st March 2022\t1 \$ = 76. 303 ₹\t\$ ₹ rate for  $21/03/2022\nSunday 20th March 2022\t1 $ = 75.998 ₹\t$ ₹ rate for <math>20/03/2022\nSatu$ rday 19th March 2022 $\t1$  \$ = 75.947 ' STEP 2:- FINDING CURRENCY'S INCLUDING THE AMOUNT IN TEXT In [39]:  $x=re.findall(r''(\d^*\.?\d^+?[\$\xi])'', data)$  $x1=re.findall(r"([$\fille{t}] ?\d*\.?\d+)", data)$ In [40]: print(x+x1) ['1 \$', '75.957 ₹', '1 \$', '75.953 ₹', '1 \$', '75.910 ₹', '1 \$', '75.825 ₹', '1 \$', '75.667 ₹', '1 \$', '76.048 ₹', '1 \$', '76.287 ₹', '1 \$', '76.275 ₹', '1 \$', '76.276 ₹', '1 \$', '76.318 ₹', '1 \$', '7 6.545 ₹', '1 \$', '76.112 ₹', '1 \$', '76.303 ₹', '1 \$', '75.998 ₹', '1 \$', '75.947 ₹', '1 \$', '75.947 ₹', '1 \$', '75.966 ₹', '1 \$', '76.020 ₹', '1 \$', '76.260 ₹', '1 \$', '76.491 ₹', '1 \$', '76.753 ₹', '1 \$', '76.756 ₹', '1 \$', '76.759 ₹', '1 \$', '76.293 ₹', '1 \$', '76.106 ₹', '1 \$', '76.901 ₹', '1 \$', '7 7.067 ₹', '1 \$', '76.439 ₹', '1 \$', '76.416 ₹', '1 \$', '76.428 ₹', '1 \$', '75.877 ₹', '1 \$', '75.593 ₹', '1 \$', '75.782 ₹', '1 \$', '75.289 ₹', '1 \$', '75.035 ₹', '1 \$', '75.073 ₹', '1 \$', '75.053 ₹', '1 \$', '75.414 ₹', '1 \$', '74.663 ₹', '1 \$', '74.643 ₹', '1 \$', '74.822 ₹', '1 \$', '74.694 ₹', '1 \$', '7 4.685 ₹', '1 \$', '74.685 ₹', '1 \$', '75.090 ₹', '1 \$', '74.972 ₹', '1 \$', '75.199 ₹', '1 \$', '75.687 ₹', '1 \$', '75.448 ₹', '1 \$', '75.640 ₹', '1 \$', '75.643 ₹', '1 \$', '75.558 ₹', '1 \$', '74.811 ₹', '1 \$', '74.689 ₹', '1 \$', '74.649 ₹', '1 \$', '74.650 ₹', '1 \$', '74.645 ₹', '1 \$', '74.644 ₹', '1 \$', '7 4.651 ₹', '1 \$', '74.820 ₹', '1 \$', '74.763 ₹', '1 \$', '74.565 ₹', '1 \$', '75.033 ₹', '1 \$', '75.002 ₹', '1 \$', '75.000 ₹', '1 \$', '75.214 ₹', '1 \$', '75.029 ₹', '1 \$', '74.755 ₹', '1 \$', '74.640 ₹', '1 \$', '74.414 ₹', '1 \$', '74.423 ₹', '1 \$', '74.423 ₹', '1 \$', '74.437 ₹', '1 \$', '74.460 ₹', '1 \$', '7 4.617 ₹', '1 \$', '74.247 ₹', '1 \$', '74.270 ₹', '1 \$', '74.389 ₹', '1 \$', '74.159 ₹', '1 \$', '73.972 ₹', '1 \$', '73.810 ₹', '1 \$', '73.821 ₹', '1 \$', '74.056 ₹', '1 \$', '74.249 ₹', '1 \$', '74.232 ₹', '1 \$', '74.465 ₹', '1 \$', '74.424 ₹', '1 \$', '74.430 ₹', '1 \$', '74.529 ₹', '1 \$', '74.396 ₹', '1 \$', '7 4.505 ₹', '1 \$', '74.513 ₹', '1 \$', '74.511 ₹', '1 \$', '74.438 ₹', '1 \$', '74.558 ₹', '1 \$', '74.721 ₹', '1 \$', '74.980 ₹', '1 \$', '75.397 ₹', '1 \$', '75.118 ₹', '1 \$', '75.115 ₹', '1 \$', '75.066 ₹', '1 \$', '75.482 ₹', '1 \$', '75.685 ₹', '1 \$', '75.744 ₹', '1 \$', '76.017 ₹', '1 \$', '76.049 ₹', '1 \$', '7 6.313 ₹', '1 \$', '76.169 ₹', '1 \$', '76.230 ₹', '1 \$', '76.056 ₹', '1 \$', '75.827 ₹', '1 \$', '75.739 ₹', '1 \$', '75.717 ₹', '1 \$', '75.792 ₹', '1 \$', '75.593 ₹', '1 \$', '75.382 ₹', '1 \$', '75.407 ₹', '1 \$', '75.392 ₹', '1 \$', '75.266 ₹', '1 \$', '75.240 ₹', '1 \$', '75.315 ₹', '1 \$', '74.998 ₹', '1 \$', '7 5.023 ₹', '1 \$', '74.997 ₹', '1 \$', '75.068 ₹', '1 \$', '75.044 ₹', '1 \$', '75.051 ₹', '1 \$', '75.056 ₹', '1 \$', '74.541 ₹', '1 \$', '74.622 ₹', '1 \$', '74.439 ₹', '1 \$', '74.436 ₹', '1 \$', '74.312 ₹', '1 \$', '74.311 ₹', '1 \$', '74.310 ₹', '1 \$', '74.186 ₹', '1 \$', '74.290 ₹', '1 \$', '74.454 ₹', '1 \$', '7 4.414 ₹', '1 \$', '74.348 ₹', '1 \$', '74.357 ₹', '1 \$', '74.346 ₹', '1 \$', '74.332 ₹', '1 \$', '74.405 ₹', '1 \$', '74.165 ₹', '1 \$', '73.918 ₹', '1 \$', '74.196 ₹', '1 \$', '74.191 ₹', '1 \$', '74.191 ₹', '1 \$', '74.477 ₹', '1 \$', '74.423 ₹', '1 \$', '74.698 ₹', '1 \$', '74.848 ₹', '1 \$', '74.935 ₹', '1 \$', '7 4.929 ₹', '1 \$', '74.929 ₹', '1 \$', '74.778 ₹', '1 \$', '75.062 ₹', '1 \$', '74.889 ₹', '1 \$', '75.096 ₹', '1 \$', '74.991 ₹', '1 \$', '75.002 ₹', '1 \$', '74.999 ₹', '1 \$', '74.861 ₹', '1 \$', '74.831 ₹', '1 \$', '75.114 ₹', '1 \$', '75.234 ₹', '1 \$', '75.023 ₹', '1 \$', '74.938 ₹', '1 \$', '75.035 ₹', '1 \$', '7 5.007 ₹', '1 \$', '75.308 ₹', '1 \$', '75.490 ₹', '1 \$', '75.420 ₹', '1 \$', '75.133 ₹', '1 \$', '75.134 ₹', '1 \$', '75.309 ₹', '1 \$', '74.854 ₹', '1 \$', '74.768 ₹', '1 \$', '74.567 ₹', '£1', '₹75.273', '£1', '₹75.3046', '£1', '₹75.2912', '£1', '₹74.7711', '£1', '₹74.1219', '£1', '₹73.4204', '£1', '₹72. 9117', '£1', '₹72.9769', '£1', '₹73.0007', '£1', '₹73.0041', '£1', '₹73.1103', '£1', '₹73.1562', '£1', '₹73.7407', '£1', '₹74.2669', '£1', '₹74.4346', '£1', '₹74.4755', '£1', '₹74.4174', '£1', '₹74. 4657', '£1', '₹74.865', '£1', '₹74.8082', '£1', '₹74.6817', '£1', '₹74.7032', '£1', '₹74.579', '£1', '₹74.6002', '£1', '₹74.5771', '£1', '₹74.302', '£1', '₹74.8179', '£1', '₹74.9475', '£1', '₹74.3339', '£1', '₹73.9089', '£1', '₹73.9155'] FINDING ONLY THE SYMBOLS OF CURRENCY IN TEXT In [41]: | curr=re.findall("([\$₹£])", data) In [42]: print("Total Number Of Currency Symbols In the TEXT DATA are: ",len(curr)) print(f"Types Of Currency Symbols In the TEXT DATA are : ",len(set(curr))," ",set(curr)) Total Number Of Currency Symbols In the TEXT DATA are : 782 Types Of Currency Symbols In the TEXT DATA are : 3 {'\$', '£', '₹'} STEP 3:-PRINTING ALL THE FORMATES OF DATES IN THE TEXT In [43]:  $|dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(d{4}))b)$ ", data) print("The Number of dates in the format of 'mm/dd/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(d{4}))b)",data)$ print("The Number of dates in the format of 'dd/mm/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d\{2\})\b)",data)$ print("The Number of dates in the format of 'dd/mm/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' print()  $dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d\{2\})\b)", data)$ print("The Number of dates in the format of 'mm/dd/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print() The Number of dates in the format of 'mm/dd/yyyy' are : 84 02/04/2022 01/04/2022 03/22/2022 03/17/2022 12/03/2022 11/03/2022 10/03/2022 09/03/2022 08/0 3/2022 07/03/2022 06/03/2022 05/03/2022 04/03/2022 03/03/2022 02/03/2022 01/03/2022 12/02/202  $2 \quad 11/02/2022 \quad 10/02/2022 \quad 09/02/2022 \quad 08/02/2022 \quad 07/02/2022 \quad 06/02/2022 \quad 05/02/2022 \quad 04/02/2022 \quad 08/02/2022 \quad$ 3/02/2022 02/02/2022 01/02/2022 12/01/2022 11/01/2022 10/01/2022 09/01/2022 08/01/2022 07/01/ 2022 06/01/2022 05/01/2022 04/01/2022 03/01/2022 02/01/2022 01/01/2022 12/12/2021 11/12/2021 10/12/2021 09/12/2021 08/12/2021 07/12/2021 06/12/2021 05/12/2021 04/12/2021 03/12/2021 02/1 2/2021 01/12/2021 12/11/2021 11/11/2021 10/11/2021 09/11/2021 08/11/2021 07/11/2021 06/11/202  $1 \quad 05/11/2021 \quad 04/11/2021 \quad 03/11/2021 \quad 02/11/2021 \quad 01/11/2021 \quad 12/10/2021 \quad 11/10/2021 \quad 10/10/2021 \quad 0$ 9/10/2021 08/10/2021 07/10/2021 06/10/2021 05/10/2021 01/01/2010 02/01/2010 03/01/2010 04/01/ 2010 05/01/2010 06/01/2010 07/01/2010 08/01/2010 09/01/2010 10/01/2010 01/11/2010 12/01/2010 The Number of dates in the format of 'dd/mm/yyyy' are : 209 02/04/2022 01/04/2022 31/03/2022 30/03/2022 29/03/2022 28/03/2022 27/03/2022 26/03/2022 25/0 3/2022 24/03/2022 23/03/2022 21/03/2022 20/03/2022 19/03/2022 18/03/2022 16/03/2022 15/03/202  $2 \quad 14/03/2022 \quad 13/03/2022 \quad 12/03/2022 \quad 11/03/2022 \quad 10/03/2022 \quad 09/03/2022 \quad 08/03/2022 \quad 07/03/2022 \quad 0$ 6/03/2022 05/03/2022 04/03/2022 03/03/2022 02/03/2022 01/03/2022 28/02/2022 27/02/2022 26/02/ 2022 25/02/2022 24/02/2022 23/02/2022 22/02/2022 21/02/2022 20/02/2022 19/02/2022 18/02/2022 17/02/2022 16/02/2022 15/02/2022 14/02/2022 13/02/2022 12/02/2022 11/02/2022 10/02/2022 09/0 2/2022 08/02/2022 07/02/2022 06/02/2022 05/02/2022 04/02/2022 03/02/2022 02/02/2022 01/02/202 2 31/01/2022 30/01/2022 29/01/2022 28/01/2022 27/01/2022 26/01/2022 25/01/2022 24/01/2022 2 3/01/2022 22/01/2022 21/01/2022 20/01/2022 19/01/2022 18/01/2022 17/01/2022 16/01/2022 15/01/ 2022 14/01/2022 13/01/2022 12/01/2022 11/01/2022 10/01/2022 09/01/2022 08/01/2022 07/01/2022 06/01/2022 05/01/2022 04/01/2022 03/01/2022 02/01/2022 01/01/2022 31/12/2021 30/12/2021 29/1 2/2021 28/12/2021 27/12/2021 26/12/2021 25/12/2021 24/12/2021 23/12/2021 22/12/2021 21/12/202 20/12/2021 19/12/2021 18/12/2021 17/12/2021 16/12/2021 15/12/2021 14/12/2021 13/12/2021 1 2/12/2021 11/12/2021 10/12/2021 09/12/2021 08/12/2021 07/12/2021 06/12/2021 05/12/2021 04/12/ 2021 03/12/2021 02/12/2021 01/12/2021 30/11/2021 29/11/2021 28/11/2021 27/11/2021 26/11/2021 25/11/2021 24/11/2021 23/11/2021 22/11/2021 21/11/2021 20/11/2021 19/11/2021 18/11/2021 17/1 1/2021 16/11/2021 15/11/2021 14/11/2021 13/11/2021 12/11/2021 11/11/2021 10/11/2021 09/11/202 1 08/11/2021 07/11/2021 06/11/2021 05/11/2021 04/11/2021 03/11/2021 02/11/2021 01/11/2021 3 1/10/2021 30/10/2021 29/10/2021 28/10/2021 27/10/2021 26/10/2021 25/10/2021 24/10/2021 23/10/ 2021 22/10/2021 21/10/2021 20/10/2021 19/10/2021 18/10/2021 17/10/2021 16/10/2021 15/10/2021 14/10/2021 13/10/2021 12/10/2021 11/10/2021 10/10/2021 09/10/2021 08/10/2021 07/10/2021 06/1 0/2021 05/10/2021 01/01/2010 02/01/2010 03/01/2010 04/01/2010 05/01/2010 06/01/2010 07/01/201 0 08/01/2010 09/01/2010 10/01/2010 01/11/2010 12/01/2010 13/01/2010 14/01/2010 15/01/2010 1 6/01/2010 17/01/2010 18/01/2010 19/01/2010 20/01/2010 21/01/2010 22/01/2010 23/01/2010 24/01/ 2010 25/01/2010 26/01/2010 27/01/2010 28/01/2010 29/01/2010 30/01/2010 31/01/2010 The Number of dates in the format of 'dd/mm/yy' are: 0 The Number of dates in the format of 'mm/dd/yy' are : 0 STEP 4 :- PRINTING ALL CARDINILITIES AND ORDERS FROM THE TEXT In [44]: | order=[]  $x=re.findall(r"((first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))|(fif[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(fou[a-z]+(th|rd|st|nd))| \\ (first|second|third|sixth)|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))|(thir[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|nd))| \\ (fou[a-z]+(th|st|st|nd))| \\ (fou[a-z]+(th|st|st|nd))| \\ (fou[a-z]+(th|st|st|nd))| \\ (fou$ h|st|nd|rd) | (fi[a-z]+th) | (six[a-z]+(th|st|nd|rd)) | (sev[a-z]+(th|st|nd|rd)) | (eig[a-z]+(th|st|nd|rd)) | (n = 1) | (n ine[a-z]+(th|st|nd|rd)) | (ten[a-z]+?(th|st|nd|rd)) | (ele[a-z]+th) | (twe[a-z]+(th|st|nd|rd)) | (hun[a-z]+(th|st|nd|rd)) | (fun[a-z]+(th|st|nd|rd)) | (fust|rd)))",data) for i in range(len(x)): if(x[i][0] not in order):order.append(x[i][0]) #print(order) orders=[] x1=re.findall(r"([0-9]+(th|st|nd|rd))", data)for i in range(len(x1)): **if**(x1[i][0] **not in** orders): orders.append(x1[i][0]) #print(orders) if order or orders: print(order+orders) else: print ("No CARDINILITIES AND ORDERS FOUND IN THE TEXT") ['seventh', 'sixth', 'fifth', 'fourth', 'third', 'second', 'first', 'thirtyfirst', 'thirtirth', 'twen tyninth', '2nd', '1st', '31st', '30th', '29th', '28th', '27th', '26th', '25th', '24th', '23rd', '22n d', '21st', '20th', '19th', '18th', '17th', '16th', '15th', '14th', '13th', '12th', '11th', '10th', '9th', '8th', '7th', '6th', '5th', '4th', '3rd', '23th', '22th', '21th'] STEP 5 :- PRINTING ALL 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT In [45]: | four=re.findall(r"(\b(a|e|i|o|u|A|E|I|O|U)[a-zA-Z]{3}\b)", data) fours=[] for i in range(len(four)): fours.append(four[i][0]) print(fours) print("Tolal number of words without repetition are : ",len(set(fours)),"\n",set(fours)) print ("NO 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT") NO 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT DATA 5 STEP 1 :- Opening File DATA 5 This TEXT file contains a book about indian history. In [46]: f=open('data5.txt') In [47]: with open('data5.txt','r',encoding='utf-8') as f: data=f.read() STEP 2:- FINDING CURRENCY'S INCLUDING THE AMOUNT IN TEXT  $x=re.findall(r"(\d^*\.?\d^+?[\$\xi])",data)$ In [48]:  $x1=re.findall(r"([$\fille{t}] ?\d*\.?\d+)", data)$ In [49]: print(x+x1) ['₹12800', '₹8190'] STEP 3 :-PRINTING ALL THE FORMATES OF DATES IN THE TEXT In [50]: | dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d{4})\b)", data) print("The Number of dates in the format of 'mm/dd/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' print()  $dates=re.findall(r"((0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d{4})\b)",data)$ print("The Number of dates in the format of 'dd/mm/yyyy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print() print("The Number of dates in the format of 'dd/mm/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print()  $dates=re.findall(r"((0[1-9]|1[0-2])/(0[1-9]|[12][0-9]|3[01])/(\d\{2\})\b)",data)$ print("The Number of dates in the format of 'mm/dd/yy' are : ",len(dates)) for i in range(len(dates)): print(dates[i][0],end=' ') print() The Number of dates in the format of 'mm/dd/yyyy' are : 10  $07/23/1916 \quad 01/09/1915 \quad 04/13/1919 \quad 06/09/1920 \quad 01/08/1920 \quad 05/02/1922 \quad 02/12/1922 \quad 02/10/1939 \quad 05/02/1922 \quad 02/10/1922 \quad 02/10/1939 \quad 05/02/1922 \quad 02/10/1922 \quad 02/$ 2/1924 06/11/1924 The Number of dates in the format of 'dd/mm/yyyy' are : 10  $01/09/1915 \quad 06/09/1920 \quad 01/08/1920 \quad 17/11/1921 \quad 05/02/1922 \quad 02/12/1922 \quad 02/10/1939 \quad 30/03/1924 \quad 05/02/1922 \quad 02/10/1939 \quad 01/08/1924 \quad 05/02/1922 \quad 01/08/1920 \quad 01/$ 2/1924 06/11/1924 The Number of dates in the format of 'dd/mm/yy' are: 0 The Number of dates in the format of 'mm/dd/yy' are: 0 STEP 4:- PRINTING ALL CARDINILITIES AND ORDERS FROM THE TEXT

<pre>STEP 5:- PRINTING ALL 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT  four=re.findall(r"(\b(a e i o u A E I O U)[a-zA-Z]{3}\b)",data) fours=[] if four:     for i in range(len(four)):         fours.append(four[i][0])     print("Tolal number of words without repetition are : ",len(set(fours)),"\n",set(fours)) else:     print("NO 4 LETTER WORDS STARTING WITH VOWELS FROM THE TEXT")  Tolal number of words without repetition are : 169 {'Indo', 'ends', 'irst', 'Afif', 'ayan', 'only', 'ezed', 'epic', 'east', 'Atop', 'eyes', 'Idea' ms', 'Adam', 'Idul', 'atop', 'Only', 'Ezra', 'imam', 'esem', 'Even', 'aced', 'ezzr', 'azzr', 'al 'Acts', 'erac', 'Azeb', 'oath', 'Usha', 'Ukil', 'atom', 'Oudh', 'Iraq', 'atma', 'atas', 'Univ',</pre>		
<pre>n', 'Imam', 'able', 'ar 'othe', 'ilea', 'amir', d', 'Isle', 'idol', 'ai 'Alai', 'Asia', 'evil', l', 'Udai', 'army', 'Im 'Upon', 'arac', 'Amma', p', 'Omar', 'Ares', 'At 'azcb', 'onsg', 'Also', t', 'Oral', 'area', 'Ax 'anew', 'Amir', 'Amar',</pre>	cotton, oshid, decom, oddin, liddy, demain, and comms', 'Azad', 'axes', 'asas', 'into', 'over', 'ease', 'Indu', 'afap', 'ence', 'Akal', 'Abla', 'Atma', 'ever', 'urge', 'Asazz', 'ones', 'Oxac', 'Arya', 'idea', 'Ahom', 'upon', 'akin', 'Aziz', 'Azha', 'Abor', 'ound', 'azxf', 'AZAD', 'azgc', 'aumad', 'onto', 'arch', 'Asaf', 'esez', 'Over', 'else', 'onIY', 'Once', 'ered', 'Azam', 'Abul', 'azzz', 'Urdu', 'Asit', 'actom', 'idal', 'Udio', 'amla', 'iron', 'Arab', 'acts', 'open', 'Alam', 'iqta', 'item', 'Each', 'used', 'each', 'Ajit', 'Actom', 'Idrs', 'awaY', 'ofto', 'ugly', 'edge', 'Iran', 'alas', 'Ajoy', 'once', 'alia', 'easy', 'URDU', 'Army', 'Aims', 'azax', 'idle', 'Oval', 'EIIu'}	'azxz', 'upI: sif', 'Adil', 'away', 'axle unt', 'ices', 'arts', 'iles cme', 'Used', 'IUCN', 'unit dha', 'azrz', 'anti', 'Anga