In [1]:
```
f = open(r'C:\Users\User\Downloads\onlinefile.txt', 'r')
p = f.read()
print(p)
f.close()
```

```
1Aaa3.5Maths2Bbb4.2Physics3Ccc7.62Chemistry4Ddd9.55Biology5Eee4.0Social6Fff7.6Englis
h7Ggg3.111Maths8Hhh9.99Physics9Iii1.23Civics
```

In [5]:
```
import pandas as pd
df = pd.read_csv(r'C:\Users\User\Downloads\onlinefile.txt')
df.to_csv('Filename2.csv',index = None)
```

In [7]:
```
import pandas as pd
df = pd.read_csv(r'C:\Users\User\Cognizance\Task_1\Filename2.csv')
print(df)
```

```
   1  Aaa    3.5      Maths
0  2  Bbb  4.200    Physics
1  3  Ccc  7.620  Chemistry
2  4  Ddd  9.550    Biology
3  5  Eee  4.000     Social
4  6  Fff  7.600    English
5  7  Ggg  3.111      Maths
6  8  Hhh  9.990    Physics
7  9  Iii  1.230     Civics
```

In [ ]:
```

```

In [7]:
```
import pandas as pd
import numpy as np

missing_values = ["NAN", "NA", "-", "Nil"]
df = pd.read_csv("https://raw.githubusercontent.com/cognizance-amrita/AI-Tasks/main/
# Number of missing values in each column
print(df.isnull().sum())
# Total number of missing values(all columns combined)
print("Total number of missing values: ",df.isnull().sum().sum())
```

```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage     14
LotArea          0
Street           0
Alley           93
LotShape         0
LandContour      0
Utilities        0
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
BldgType         0
HouseStyle       0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
RoofStyle        0
RoofMatl         0
Exterior1st      0
Exterior2nd      0
MasVnrType       0
MasVnrArea       0
ExterQual        0
ExterCond        0
Foundation       0
BsmtQual         3
BsmtCond         3
BsmtExposure     3
BsmtFinType1     3
BsmtFinSF1       0
BsmtFinType2     3
dtype: int64
Total number of missing values:  122
```

In [14]:
```
# Replacing the missing values according to each column
import pandas as pd
import numpy as np

missing_values = ["NAN", "NA", "-", "Nil"]
df = pd.read_csv("https://raw.githubusercontent.com/cognizance-amrita/AI-Tasks/main/
df['LotFrontage'].fillna(50, inplace=True)
df['Alley'].fillna('Pave', inplace=True)
df['BsmtQual'].fillna('Gd', inplace=True)
df['BsmtCond'].fillna('Gd', inplace=True)
df['BsmtExposure'].fillna('Mn', inplace=True)
df['BsmtFinType1'].fillna('ALQ', inplace=True)
```

```
df['BsmtFinType2'].fillna('Unf', inplace=True)
print(df.isnull().sum()) # All NIL values are filled
```

```
Id              0
MSSubClass      0
MSZoning        0
LotFrontage     0
LotArea         0
Street          0
Alley           0
LotShape        0
LandContour     0
Utilities       0
LotConfig       0
LandSlope       0
Neighborhood    0
Condition1      0
Condition2      0
BldgType        0
HouseStyle      0
OverallQual     0
OverallCond     0
YearBuilt       0
YearRemodAdd    0
RoofStyle       0
RoofMatl        0
Exterior1st     0
Exterior2nd     0
MasVnrType      0
MasVnrArea      0
ExterQual       0
ExterCond       0
Foundation      0
BsmtQual        0
BsmtCond        0
BsmtExposure    0
BsmtFinType1    0
BsmtFinSF1      0
BsmtFinType2    0
dtype: int64
```

In [ ]:

In [81]:
```
# Reading file
f = open(r'C:\Users\User\Downloads\about.txt','r')
p = f.read()
print(p)
f.close()
```

```
Python has tools for almost every aspect of scientific computing. The Bank of Americ
a uses Python to crunch its financial data and Facebook looks upon the Python librar
y Pandas for its data analysis. While there are many libraries available to perform
data analysis in Python, here are a few: NumPy, SciPy, Pandas and Matplotlib.
```

In [68]:
```
# Words with atleast 6 letters
with open(r'C:\Users\User\Downloads\about.txt','r') as p:
    for i in p:
        j = i.split(".")
        for k in j:
            s = k.split(" ")
            for h in s:
                if len(h)>=6:
                    if h[-1]!=',':
                        print(h)
                    elif len(h)>6 and h[-1] == ',':
                        print(h[0:6])
```

```
Python
almost
aspect
scientific
computing
America
Python
crunch
financial
Facebook
Python
library
Pandas
analysis
libraries
available
perform
analysis
Python
Pandas
Matplotlib
```

In [82]:
```
# Most frequently used word
f = open(r'C:\Users\User\Downloads\about.txt','r')
f1 = ""
c = 0
l = []

for i in f:
    j = i.lower().replace(',','').replace('.','').split(" ");
    for w in j:
        l.append(w);

for i in range(0, len(l)):
    k = 1;
    for j in range(i+1, len(l)):
        if(l[i] == l[j]):
```

```
            k = k + 1;

    if(k > c):
        c = k;
        f1 = l[i];

print("Most frequently used word: " + f1)
print("count: " + str(c))
f.close();
```

```
Most frequently used word: python
count: 4
```

In [ ]:

In [35]:
```
# Sorted array
import pandas as pd
import requests
import io
import numpy

url = "https://raw.githubusercontent.com/cognizance-amrita/AI-Tasks/main/Q4-Dataset.
download = requests.get(url).content
df = pd.read_csv(io.StringIO(download.decode('utf-8')))
arr = df.to_numpy()
arr = arr[arr[:, 2].argsort()]
print(arr)
```

```
[['B005P0HHGK' 170 0.1]
 ['B000JEHAHS' 133 0.12]
 ['B0001PB9FY' 183 0.19]
 ['B005DUM9UQ' 138 0.24]
 ['B002GWHC0G' 165 0.26]
 ['B0001PB9FE' 148 0.29]
 ['B0093NIWVO' 127 0.31]
 ['B000G6RPMY' 111 0.38]
 ['B0017I8UME' 139 0.43]
 ['B0019CW0HE' 178 0.43]
 ['B0017129DC' 159 0.45]
 ['B000E7L2R4' 148 0.48]
 ['B003SO503C' 162 0.6]
 ['B001GVISJM' 177 0.61]
 ['B001EO5ZME' 156 0.64]
 ['B0064KO0BU' 157 0.74]
 ['B001KUUNP6' 178 0.74]
 ['B0025WIAN0' 82 0.77]
 ['B000GGKQSO' 185 0.83]
 ['B000ITVLE2' 126 0.84]
 ['B000LKZK7C' 194 0.88]
 ['B00171APVA' 184 1.0]
 ['B001EO5ZMO' 178 1.04]
 ['B000NY8ODS' 162 1.08]
 ['B000WFRMRW' 174 1.13]
 ['B0064KU9HO' 159 1.15]
 ['B004OYBN7C' 81 1.29]
 ['B001L4ELRW' 108 1.53]
 ['B003ZFRKGO' 191 1.59]
 ['B0036VM05I' 174 1.72]
 ['B006K2ZZ7K' 157 1.76]
 ['B003VTN95K' 197 1.84]
 ['B00144C10S' 182 1.86]
 ['B002MV23XM' 148 1.89]
 ['B0066DMI6Y' 148 1.9]
 ['B001SATU8E' 166 1.91]
 ['B000E7VI7S' 103 1.99]
 ['B001GVISJW' 144 2.05]
 ['B0009XLVGA' 195 2.11]
 ['B001EO5TPM' 161 2.11]
 ['B007TFONH0' 152 2.15]
 ['B0040WAG7Q' 90 2.2]
 ['B0028C44Z0' 151 2.22]
 ['B001FB69YY' 98 2.24]
 ['B0048IC328' 88 2.29]
 ['B007J32WX4' 104 2.29]
 ['B0009XLVG0' 116 2.36]
 ['B001REEG6C' 100 2.44]
 ['B001GVISJC' 81 2.47]
```

```
['B00029XIZI' 81 2.59]
['B002X9JNYU' 171 2.61]
['B001EO5ZMY' 114 2.66]
['B000J0HIT2' 180 2.7]
['B003F6UO7K' 141 2.7]
['B004X2KR36' 126 2.74]
['B004K2IHUO' 175 2.82]
['B007JFV6RK' 197 2.89]
['B00283TPYE' 100 2.94]
['B001EPQ0J0' 150 3.09]
['B000SV90J8' 118 3.22]
['B004N5KULM' 166 3.23]
['B0026Y3YBK' 180 3.26]
['B001E4KFG0' 140 3.43]
['B00821UN4M' 153 3.47]
['B003SE19UK' 160 3.48]
['B00473RWXY' 150 3.48]
['B0081XN2HQ' 129 3.51]
['B000H13270' 80 3.52]
['B0002567IW' 163 3.53]
['B000UA0QIQ' 161 3.64]
['B002HQAXUW' 165 3.68]
['B0059WXJKM' 148 3.73]
['B001EO5QW8' 165 3.74]
['B00813GRG4' 183 3.91]
['B0025VRCJY' 153 3.91]
['B001D07IPG' 197 3.93]
['B006SQBRMA' 188 3.97]
['B0064KOUNI' 152 3.98]
['B005CJVVJ8' 108 4.05]
['B003OB0IB8' 133 4.08]
['B003ZFXJDW' 110 4.11]
['B007B9J6G2' 80 4.18]
['B001EPPI84' 184 4.21]
['B004V6AH34' 103 4.24]
['B001HTL6CY' 160 4.26]
['B007DJ0O9I' 94 4.31]
['B005R8JE8O' 83 4.34]
['B0037LW78C' 195 4.55]
['B003TQQKFQ' 113 4.55]
['B003EMU7EU' 114 4.56]
['B001IUKD76' 89 4.58]
['B000O9Y62A' 97 4.59]
['B000LQOCH0' 160 4.6]
['B001UJEN6C' 99 4.63]
['B002SRYRE8' 189 4.67]
['B00374XSVY' 120 4.7]
['B002TDK0VK' 196 4.8]
['B0037ZFEW4' 117 4.85]
['B003YDP5PA' 169 4.98]]
```

In [36]:

```
# Sorted table
import pandas as pd
import requests
import io
import numpy

url = "https://raw.githubusercontent.com/cognizance-amrita/AI-Tasks/main/Q4-Dataset.
download = requests.get(url).content
df = pd.read_csv(io.StringIO(download.decode('utf-8')))
arr = df.to_numpy()
arr = arr[arr[:, 2].argsort()]
```

```
df1 = pd.DataFrame(arr, columns=['ProductID', 'price', 'rating'])
print(df1)
```

```
     ProductID price rating
0   B005P0HHGK   170    0.1
1   B000JEHAHS   133   0.12
2   B0001PB9FY   183   0.19
3   B005DUM9UQ   138   0.24
4   B002GWHC0G   165   0.26
..         ...   ...    ...
94  B002SRYRE8   189   4.67
95  B00374XSVY   120    4.7
96  B002TDK0VK   196    4.8
97  B0037ZFEW4   117   4.85
98  B003YDP5PA   169   4.98

[99 rows x 3 columns]
```

In [ ]:

In [8]:

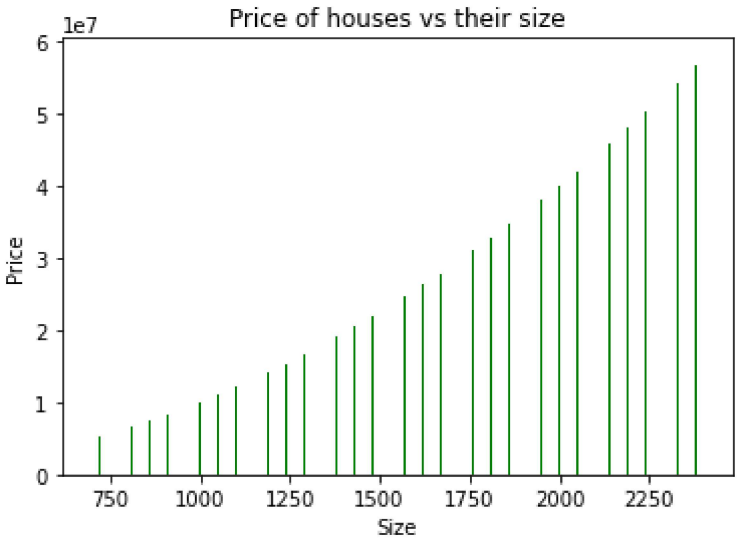
```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv(r'C:\Users\User\Downloads\Q5.csv')

df = pd.DataFrame(data)

X = list(df.iloc[:, 0])
Y = list(df.iloc[:, 1])

plt.bar(X, Y, color='g')
plt.title("Price of houses vs their size")
plt.xlabel("Size")
plt.ylabel("Price")
plt.show()
```



x	y
700	4901400
710	5042420
720	5185440
730	5330460
740	5477480
750	5626500
760	5777520
770	5930540
780	6085560
790	6242580
800	6401600
810	6562620
820	6725640
830	6890660
840	7057680
850	7226700
860	7397720
870	7570740
880	7745760
890	7922780
900	8101800
910	8282820
920	8465840
930	8650860
940	8837880
950	9026900
960	9217920
970	9410940
980	9605960
990	9802980
1000	10002000
1010	10203020
1020	10406040
1030	10611060
1040	10818080
1050	11027100
1060	11238120
1070	11451140
1080	11666160
1090	11883180
1100	12102200
1110	12323220
1120	12546240
1130	12771260
1140	12998280
1150	13227300
1160	13458320
1170	13691340
1180	13926360

1190 14163380
1200 14402400
1210 14643420
1220 14886440
1230 15131460
1240 15378480
1250 15627500
1260 15878520
1270 16131540
1280 16386560
1290 16643580
1300 16902600
1310 17163620
1320 17426640
1330 17691660
1340 17958680
1350 18227700
1360 18498720
1370 18771740
1380 19046760
1390 19323780
1400 19602800
1410 19883820
1420 20166840
1430 20451860
1440 20738880
1450 21027900
1460 21318920
1470 21611940
1480 21906960
1490 22203980
1500 22503000
1510 22804020
1520 23107040
1530 23412060
1540 23719080
1550 24028100
1560 24339120
1570 24652140
1580 24967160
1590 25284180
1600 25603200
1610 25924220
1620 26247240
1630 26572260
1640 26899280
1650 27228300
1660 27559320
1670 27892340
1680 28227360

1690 28564380
1700 28903400
1710 29244420
1720 29587440
1730 29932460
1740 30279480
1750 30628500
1760 30979520
1770 31332540
1780 31687560
1790 32044580
1800 32403600
1810 32764620
1820 33127640
1830 33492660
1840 33859680
1850 34228700
1860 34599720
1870 34972740
1880 35347760
1890 35724780
1900 36103800
1910 36484820
1920 36867840
1930 37252860
1940 37639880
1950 38028900
1960 38419920
1970 38812940
1980 39207960
1990 39604980
2000 40004000
2010 40405020
2020 40808040
2030 41213060
2040 41620080
2050 42029100
2060 42440120
2070 42853140
2080 43268160
2090 43685180
2100 44104200
2110 44525220
2120 44948240
2130 45373260
2140 45800280
2150 46229300
2160 46660320
2170 47093340
2180 47528360

2190 47965380
2200 48404400
2210 48845420
2220 49288440
2230 49733460
2240 50180480
2250 50629500
2260 51080520
2270 51533540
2280 51988560
2290 52445580
2300 52904600
2310 53365620
2320 53828640
2330 54293660
2340 54760680
2350 55229700
2360 55700720
2370 56173740
2380 56648760
2390 57125780
2400 57604800