Part B

1. Write a program to copy line by line from a file to another and count the number of words, spaces, newline character, percentage of vowels and consonants in the file.

```
file_to_read="ReadData.txt"
file_to_write="WriteData.txt"

file=open(file_to_read,"r")
data=file.readlines()
file.close()

with open(file_to_write,"a") as file:
    for s in data:
        file.write(s)

with open(file_to_write) as file:
    text=file.read()
    count_space=0
    count_n1=0
    count_vowels=0
    count_consonants=0
    count_words=0

    for char in text:
        if char==' ':
            count_space+=1
        if char=='\n':
            count_n1+=1
        if char in "aeiou":
            count_vowels+=1
        if char not in "aeiou":
            count_consonants+=1
    words = text.split()
    count_words += len(words)

print("Spaces=",count_space)
print("New line=",count_n1)
print("Words=",count_words)
print("Percentage of Vowels=",int((count_vowels)*100/len(text)),"%")
print("Percentage of Consonants=",int((count_consonants)*100/len(text)),"%")
```

2. Write a program to create a text file “MyFile.txt” and write a few lines into it. Replace all the spaces from text with - (dash).

```
i=0
n=int(input("Enter number of lines to write to a file"))
with open("MyFile.txt","a") as f1:
    while i<=n:
        text = input("Enter text to append in the file:")
        f1.writelines(text)
        i=i+1
with open("MyFile.txt","r") as f1:
    data = f1.read()
    data=data.replace(' ','-')
```

```
with open("MyFile.txt","w") as f1:
    f1.write(data)
```

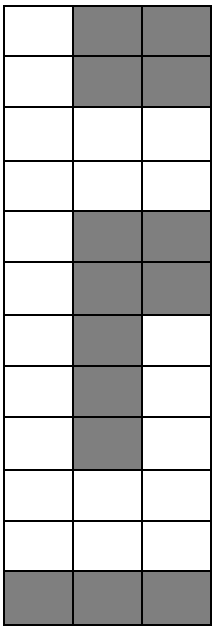
3. Using numpy, perform the following:
   - Create a 3x4 matrix filled with values from 10 to 21.

```
import numpy as np
m= np.arange(10,22).reshape((3, 4))
print(m)
```

Output:

```
[[10 11 12 13]
 [14 15 16 17]
 [18 19 20 21]]
```

   - Create a 3x3 matrix and print the elements of the highlighted matrix shown in below figures.



(a) (b) (c) (d)

```
import numpy as np
d=np.random.randint(1,9,size=(3,3))
print(d)
print("Fig.a")
print(d[:2,1:])
print("Fig.b")
print(d[1:,1:])
print("Fig.c")
print(d[:,1])
print("Fig.d")
print(d[2,:])
```

Output:

```
[[8 7 5]
 [7 2 8]
 [2 3 6]]
```
Fig.a
```
[[7 5]
 [2 8]]
```
Fig.b
```
[[2 8]
 [3 6]]
```
Fig.c
```
[7 2 3]
```
Fig.d
```
[2 3 6]
```

- Create an element-wise comparison (greater, greater_equal, less and less_equal,equal, not_eqaul) of any two arrays each having a size of 2x2x3.

```
import numpy as np
x = np.array([[[ 2,  3,  4],
  [ 5,  6,  7]],
 [[ 8,  9, 10],
  [11, 1, 13]]])
y = np.array([[[12, 13, 14],
  [15, 1, 17]],
 [[18, 19, 20],
  [21, 2, 23]]])

print("Original numbers:")
print(x)
print(y)
print("Comparison - greater")
print(np.greater(x, y))
print("Comparison - greater_equal")
print(np.greater_equal(x, y))
print("Comparison - less")
print(np.less(x, y))
print("Comparison - less_equal")
print(np.less_equal(x, y))
```

Output:

```
Original numbers:
[[[ 2  3  4]
  [ 5  6  7]]
```

```
 [[ 8  9 10]
  [11  1 13]]]
[[[12 13 14]
  [15  1 17]]

 [[18 19 20]
  [21  2 23]]]
Comparison - greater
[[[False False False]
  [False  True False]]

 [[False False False]
  [False False False]]]
Comparison - greater_equal
[[[False False False]
  [False  True False]]

 [[False False False]
  [False False False]]]
Comparison - less
[[[ True  True  True]
  [ True False  True]]

 [[ True  True  True]
  [ True  True  True]]]
Comparison - less_equal
[[[ True  True  True]
  [ True False  True]]

 [[ True  True  True]
  [ True  True  True]]]
```

- Compute sum of all elements, sum of each column and sum of each row of a given array.

```
x = np.array([[0,1],[2,3]])
print("Original array:")
print(x)
print("Sum of all elements:")
print(np.sum(x))
print("Sum of each column:")
print(np.sum(x, axis=0))
print("Sum of each row:")
print(np.sum(x, axis=1))
```

Output:

```
Original array:
```

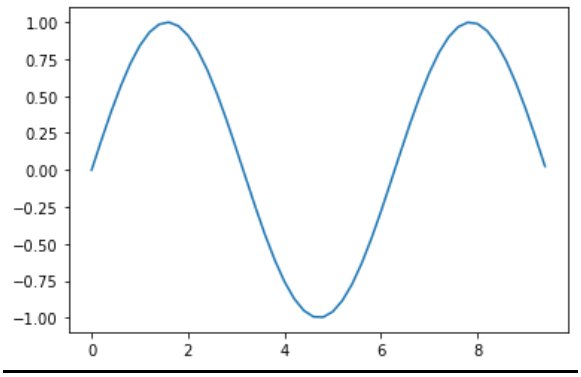
```
[[0 1]
 [2 3]]
Sum of all elements:
6
Sum of each column:
[2 4]
Sum of each row:
[1 5]
```

4. Using numpy, perform the following:
   - Compute the x and y coordinates for points on a sine curve and plots the points using matplotlib.

```
import numpy as np
import matplotlib.pyplot as plt
# Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.2)
y = np.sin(x)
print("Plot the points using matplotlib:")
plt.plot(x, y)
plt.show()
```

Output:

Plot the points using matplotlib:



   - Create a 3x4 matrix and find any missing data, number of zeros in a given array.

```
nums = np.array([[3, 0, np.nan, 1], [10, 12, 0, 9],[5, np.nan, 1, np.nan]])
print("Original array:")
print(nums)
print("\nMissing data of the said array:")
print(np.isnan(nums))
print("\nNumber of zeros in a given array:",nums.size-np.count_nonzero(nums))
```

Output:

Original array:

```
[[ 3.  0. nan  1.]
 [10. 12.  0.  9.]
 [ 5. nan  1. nan]]

Missing data of the said array:
[[False False  True False]
 [False False False False]
 [False  True False  True]]

Number of zeros in a given array: 2
```

- Create a matrix of randomly generated data of shape (4, 4) and replace all positive values with 2 and all negative values with –2.

```
arr = np.random.randn(4, 4)
print(arr)
np.where(arr > 0, 2, -2)
```

```
Output:
[[ 1.11960514  0.02707735  0.01637816  2.11162992]
 [ 0.89251876 -0.22738665  0.08530356  1.35389111]
 [-0.09071456  2.02610418  0.12614846 -1.04069065]
 [-0.97317884  1.11388901 -1.58432008 -0.18095013]]
Out[3]:
array([[ 2,  2,  2,  2],
       [ 2, -2,  2,  2],
       [-2,  2,  2, -2],
       [-2,  2, -2, -2]])
```

- Swap the rows and columns of given array in reverse order.

```
nums = np.array([[[1, 2, 3, 4], [0, 1, 3, 4], [90, 91, 93, 94], [5, 0, 3, 2]]])
print("Original array:")
print(nums)
print("\nSwap rows and columns in reverse order:")
new_nums = print(nums[::-1, ::-1])
print(new_nums)
```

Output:

```
Original array:
[[[ 1  2  3  4]
  [ 0  1  3  4]
  [90 91 93 94]
  [ 5  0  3  2]]]
```

```
Swap rows and columns in reverse order:
[[[ 5  0  3  2]
  [90 91 93 94]
  [ 0  1  3  4]
  [ 1  2  3  4]]]
```

- Find common values between two arrays.

```
randnums1= np.random.randint(1,100, size=(3,3))
print("array 1:",randnums1)
randnums2= np.random.randint(1,100, size=(3,3))
print("array 2:",randnums2)
print("Common elements are:")
print(np.intersect1d(randnums1, randnums2))
```

Output:

```
array 1: [[90 14 14]
 [77 47 36]
 [99 42 13]]
array 2: [[89 91 82]
 [37 13 95]
 [47 32  8]]
Common elements are:
[13 47]
```

5. Using Pandas, perform the following:
   - Convert a numpy array and dictionary to a Pandas series.

```
import numpy as np
import pandas as pd
np_array = np.array([10, 20, 30, 40, 50])
print("NumPy array:")
print(np_array)
new_series = pd.Series(np_array)
print("Converted array to Pandas series:")
print(new_series)
dct = {'Sunday':1,'Monday':2,'Tuesday':3,'Wednesday':4,'Thursday':5,
 'Friday':6,'Saturday':7}
new_series1 = pd.Series(dct)
print("Converted array to Pandas series:")
print(new_series1)
```

Output:

```
NumPy array:
[10 20 30 40 50]
Converted array to Pandas series:
0    10
1    20
```

```
2    30
3    40
4    50
dtype: int64
Converted array to Pandas series:
Sunday      1
Monday      2
Tuesday     3
Wednesday   4
Thursday    5
Friday      6
Saturday    7
dtype: int64
```

- Convert Series of lists to one Series and sort the values.

```
import pandas as pd
s = pd.Series([
                ['Red', 'Green', 'White'],
                ['Red', 'Black'],
                ['Yellow']])
print("Original Series of list")
print(s)
s = s.apply(pd.Series).stack().reset_index(drop=True)
print("One Series")
print(s)
new_s = pd.Series(s).sort_values()
print("After sorting")
print(new_s)
```

Output:

```
Original Series of list
0   [Red, Green, White]
1             [Red, Black]
2                  [Yellow]
dtype: object
One Series
0      Red
1    Green
2    White
3      Red
4    Black
5   Yellow
dtype: object
After sorting
4    Black
1    Green
0      Red
3      Red
```

```
2    White
5   Yellow
dtype: object
```

- Find the positions of numbers that are multiples of 5 of a given integer number series.

```
n_series = pd.Series(np.random.randint(1, 50, 9))
print("Original Series:\n")
print(n_series)
a=n_series.to_numpy()
index = np.argwhere(a % 5==0)
print("indexes of number divisible by 5")
print(index)
```

Output:

```
Original Series:
0   16
1   48
2   22
3   38
4   14
5   25
6   15
7   17
8   45
dtype: int32
indexes of number divisible by 5
[[5]
 [6]
 [8]]
```

- Find the number of occurrences of each unique value in a Series.

```
df = pd.DataFrame({"col1": ["a", "b", "a", "c", "a", "a", "a", "c"]})
print(df)
item_counts = df["col1"].value_counts()
print("Number of occurence of each unique value")
print(item_counts)
```

Output

```
col1
0   a
1   b
2   a
3   c
4   a
5   a
6   a
7   c
Number of occurence of each value
a   5
```

```
c   2
b   1
Name: col1, dtype: int64
```

- Get the positions of items of a given series in another given series.

```
series1 = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
series2 = pd.Series([1, 3, 5, 7, 10])
print("Original Series:")
print(series1)
print(series2)
result = [pd.Index(series1).get_loc(i) for i in series2]
print("Positions of items of series2 in series1:")
print(result)
```

Output

```
Original Series:
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
9   10
dtype: int64
0    1
1    3
2    5
3    7
4   10
dtype: int64
Positions of items of series2 in series1:
[0, 2, 4, 6, 9]
```

6. Using Pandas, perform the following:
   - Create and display a dataFrame from a dictionary of names of the student, subjects, grade of respective subject, number of each attempt,sore and qualify data which has the index labels.

```
exam_data  = {'name': ['Anil', 'Avinash', 'Kaveri', 'James', 'kavitha', 'Pushpa'],
              'Subjects': ["Kannada", "English", "Hindi", "Tamil", "Telagu", "Gujarathi"],
              'attempts': [1, 3, 2, 3, 2, 3, ],
              'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes'],
             'Score':[30,12,32,14,8,40]}
labels = ['a', 'b', 'c', 'd', 'e', 'f']
```

```
df = pd.DataFrame(exam_data , index=labels)
print(df)
```

Output:

```
    name   Subjects  attempts qualify  Score
a    Anil    Kannada        1       yes        30
b  Avinash    English       3       no         12
c   Kaveri     Hindi         2       yes        32
d    James     Tamil         3       no         14
e  kavitha     Telagu        2       no         8
f   Pushpa  Gujarathi       3      yes        40
```

- Display the details of a specified student.

```
df = pd.DataFrame(exam_data , index=labels)
print(df.iloc[:1])
```

output:

```
    name   Subjects  attempts qualify  Score
a    Anil    Kannada        1       yes        30
```

- Select the rows where the number of attempts in the examination is greater than 2 and score less than 40.

```
print("Number of attempts in the examination is greater than 2 and score less than 40 :")
print(df[(df['attempts'] > 2) & (df['Score'] < 40)])
```

Output:

Number of attempts in the examination is greater than 2 and scores less than 40:

```
    name   Subjects  attempts qualify  Score
b  Avinash  English        3            no     12
d   James    Tamil         3            no     14
```

- Select the rows where the number of attempts in the examination is less than 2 and score greater than 15.

```
print("Number of attempts in the examination is less than 2 and score greater than 15 :")
print(df[(df['attempts'] < 2) & (df['Score'] > 15)])
```

Output:

```
Number of attempts in the examination is less than 2 and score greater than 15 :
    name Subjects  attempts qualify  Score
a  Anil  Kannada        1           yes    30
```

- Sort the data frame first by 'name' in descending order, then by 'score' in ascending order.

```
print("'name' in descending order")
print(df.sort_values(by=['name','Score'],ascending=[True,False]))
print("'Score' in ascending order")
```

```
print(df.sort_values(by=['name','Score'],ascending=[False,True]))
```

Output:

'name' in descending order

	name	Subjects	attempts	qualify	Score
a	Anil	Kannada	1	yes	30
b	Avinash	English	3	no	12
d	James	Tamil	3	no	14
c	Kaveri	Hindi	2	yes	32
f	Pushpa	Gujarathi	3	yes	40
e	kavitha	Telagu	2	no	8

'Score' in ascending order

	name	Subjects	attempts	qualify	Score
e	kavitha	Telagu	2	no	8
f	Pushpa	Gujarathi	3	yes	40
c	Kaveri	Hindi	2	yes	32
d	James	Tamil	3	no	14
b	Avinash	English	3	no	12
a	Anil	Kannada	1	yes	30