PROGRAM 1

a) Write a python program to find the best of two test average marks out of three test's marks acceptedfrom the user.

SOURCE CODE

```
# take inputs
num1 = float(input('Enter first number: '))
num2 = float(input('Enter second number: '))
num3 = float(input('Enter third number: '))

# calculate average
avg = (num1 + num2 + num3)/3

# display result
print('The average of numbers = %0.2f' %avg)
```

OUTPUT

```
Enter first number: 10
Enter second number: 2
Enter third number: 30
The average of numbers = 14.00
```

b) Develop a Python program to check whether a given number is palindrome or not and also count thenumber of occurrences of each digit in the input number.

SOURCE CODE

```
from collections import Counter
value = input("Enter a value : ")

if value == value[::-1]:
   print("Palindrome")
else:
   print("Not Palindrome")

counted_dict = Counter(value)
for key in sorted(counted_dict.keys()):
   print(f'{key} appears {counted_dict[key]} times');
```

OUTPUT

```
Enter a value : 11
Palindrome
1 appears 2 times

Enter a value : 123332
Not Palindrome
1 appears 1 times
2 appears 2 times
3 appears 3 times
```

PROGRAM 2

A) Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

SOURCE CODE

```
def fn(n):
    if n == 1:
        return 0
    elif n == 2:
        return 1
    else:
        return fn(n-1) + fn(n-2)
num = int(input("enter a number: "))
if num > 0:
    print("fn(",num,") = ",fn(num),sep ="")
else:
    print("error in input")
```

OUTPUT

```
Enter a number : 6
fn(6) = 5

Enter a number : asc
Try with numeric value

Enter a number : -3
Input should be greater than 0
```

B) Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

SOURCE CODE

```
def BinToDec(b):
    return int(b,2)
def OctToHex(o):
    return hex(int(o,8))
bnum = input("enter the binary number: ")
dnum = BinToDec(bnum)
print("\nEquivalent Decimal value = ",dnum)

onum = input("enter the octal number: ")
hnum = OctToHex(onum)
print("\nEquivalent Hexadecimal value = ",hnum[2:].upper())
```

OUTPUT

```
Enter the binary number : 1010
Equivalent Decimal value = 10

Enter the octal number : 73
Equivalent hexadecimal value = 3B
```

PROGRAM 3

A) Write a Python program that accepts a sentence and find the number of words,digits, uppercase letters and lowercase letters.

SOURCE CODE

```
sentence = input("enter a sentence: ")
wordList = sentence.split(" ")
print("this sentence has", len(wordList), "words")

digCnt = upCnt = loCnt = 0

for ch in sentence:
    if '0' <= ch <= '9':
        digCnt += 1
    elif 'A' <= ch <= 'Z':
        upCnt += 1
    elif 'a' <= ch <= 'z':
        loCnt += 1

print("This sentence has", digCnt, "digits", upCnt, "upper case letters",
 loCnt, "lower case letters")
```

OUTPUT

```
enter a sentence :
John went to market
This sentence has 4 words
This sentence has 0 digits 1 upper case letters 16 lower case letters
```

B) Write a Python program to find the string similarity between two given strings

SOURCE CODE

```
str1 = input("Enter String 1 \n")
str2 = input("Enter String 2 \n")

if len(str2)  <  len(str1):
     short = len(str2)
     long = len(str1)
 else:
     short = len(str1)
     long = len(str2)


match_count = 0
for i in range(short_string_length):
    if str1[i] == str2[i]:
        match_count += 1

print("Similarity between two said strings: ")
print(match_count/long_string_length)
```

OUTPUT

```
Enter String 1
HAPPY
Enter String 2
GOOGLE
Similarity between two said strings:
0.0

Enter String 1 : SWEET
Enter String 2 : SWEET
Similarity between strings "SWEET" and "SWEET" is :  1.0

Enter String 1 : FACE
Enter String 2 : FACEBOOK
Similarity between strings "FACE" and "FACEBOOK" is :
0.6666666666666666
```

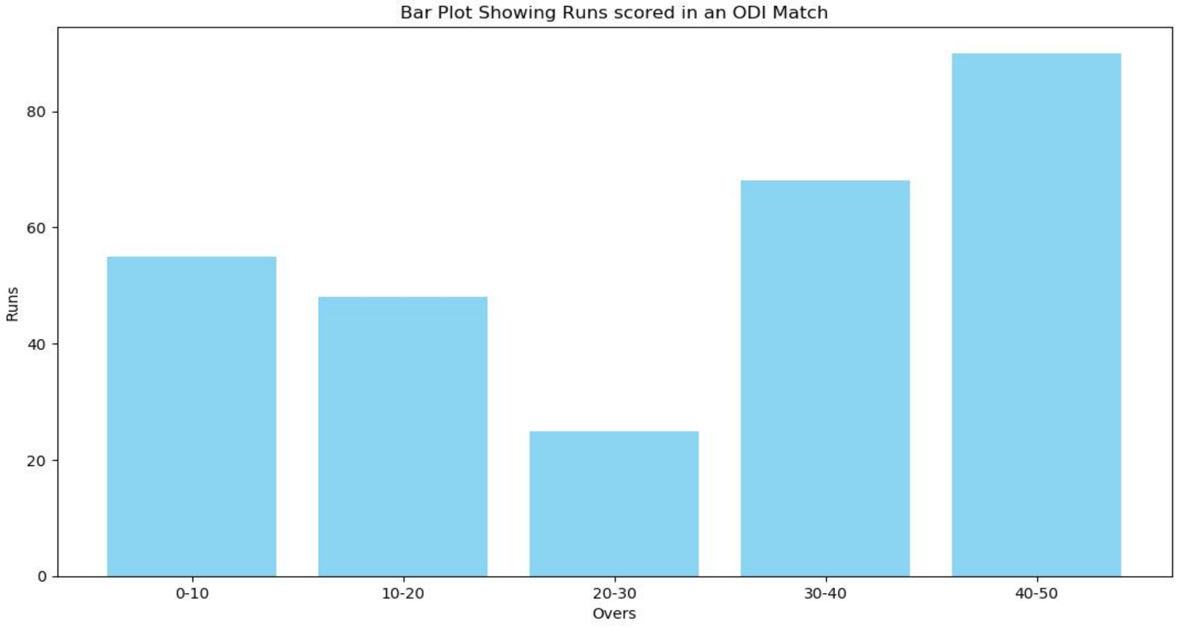
PROGRAM 4

A) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.

SOURCE CODE

```
import matplotlib.pyplot as plt

# Sample data for demonstration
categories = ['0-10', '10-20', '20-30', '30-40', '40-50']
values = [55, 48, 25, 68, 90]

# Create a bar plot
plt.bar(categories, values, color='skyblue')

# Add labels and title
plt.xlabel('Overs')
plt.ylabel('Runs')
plt.title('Bar Plot Showing Runs scored in an ODI Match')

# Display the plot
plt.show()
```

OUTPUT

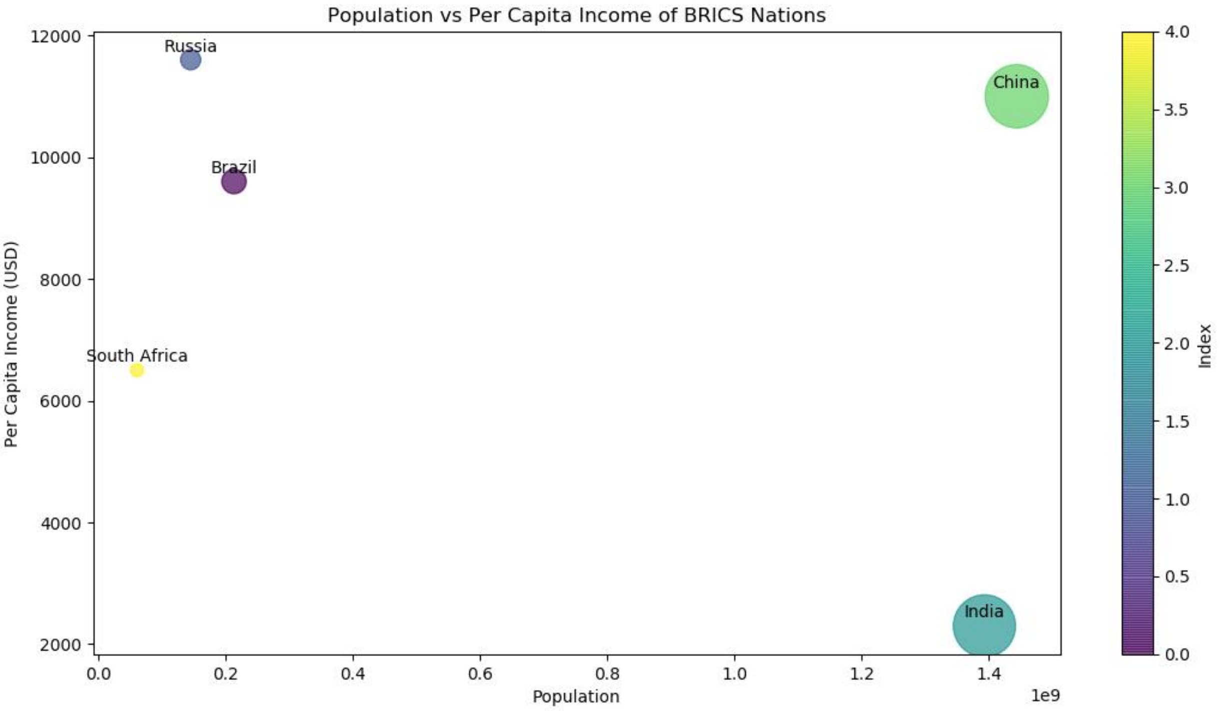


B) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib

SOURCE CODE

```
import matplotlib.pyplot as plt
import numpy as np

# BRICS nations data (hypothetical)
countries = ['Brazil', 'Russia', 'India', 'China', 'South Africa']
population = [213993437, 145912025, 1393409038, 1444216107, 61608912]

per_capital_income = [9600, 11600, 2300, 11000, 6500]
 # Per capital income in USD

# Scale the population for circle size
circle_size = [pop / 1000000 for pop in population]
# Scaling down for better visualization

# Assign different colors based on index
colors = np.arange(len(countries))

# Create a scatter plot with varying circle sizes and colors
scatter = plt.scatter(population, per_capita_income, s=circle_size, c=colors,
cmap='viridis', alpha=0.7, label='BRICS Nations')

# Annotate each point with the country name
for i, country in enumerate(countries):
plt.annotate(country, (population[i], per_capita_income[i]), textcoords="offset
points", xytext=(0,5), ha='center')

# Add colorbar
plt.colorbar(scatter, label='Index')

# Add labels and title
plt.xlabel('Population')
plt.ylabel('Per Capita Income (USD)')
plt.title('Population vs Per Capita Income of BRICS Nations')

# Display the plot
plt.show()
```

OUTPUT


Population vs Per Capita Income of BRICS Nations
Russia
China
Brazil
South Africa
India
Per Capita Income (USD)
12000
10000
8000
6000
4000
2000
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Population
1e9
Index
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0

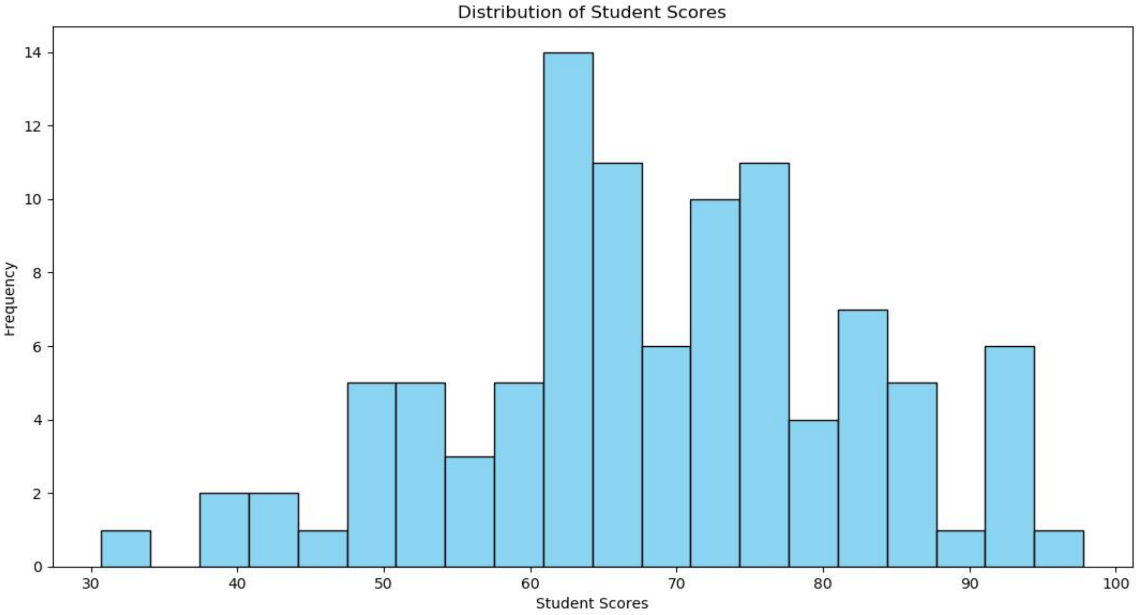
PROGRAM 5

A) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib.

SOURCE CODE

```
import matplotlib.pyplot as plt
import numpy as np

# Generate random student scores (example data)
np.random.seed(42)
student_scores = np.random.normal(loc=70, scale=15, size=100)

# Create a histogram plot
plt.hist(student_scores, bins=20, color='skyblue', edgecolor='black')

# Add labels and title
plt.xlabel('Student Scores')
plt.ylabel('Frequency')
plt.title('Distribution of Student Scores')

# Display the plot
plt.show()
```

OUTPUT

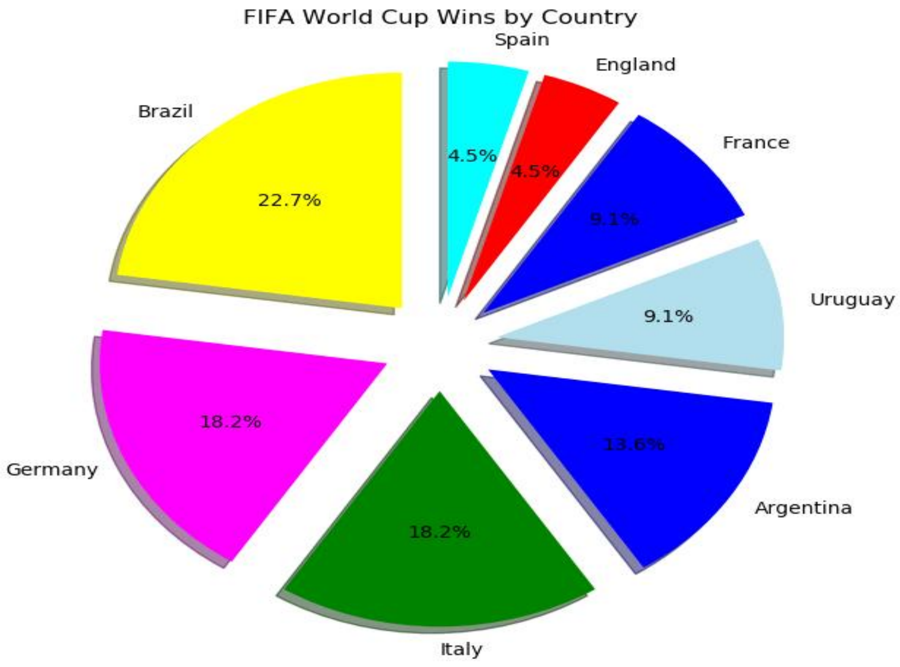

Distribution of Student Scores
Frequency
0
2
4
6
8
10
12
14
30
40
50
60
70
80
90
100
Student Scores

B) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib.

SOURCE CODE

```
import matplotlib.pyplot as plt

#Number of FIFA World Cup wins for different countries
countries = ['Brazil', 'Germany', 'Italy', 'Argentina', 'Uruguay', 'France', 'England', 'Spain']
wins = [5, 4, 4, 3, 2, 2, 1, 1]  # Replace with actual data

# Colors for each country
colors = ['yellow', 'magenta', 'green', 'blue', 'lightblue', 'blue', 'red', 'cyan']

plt.pie(wins, labels=countries, autopct='%1.1f%%', colors=colors,  startangle=90,
explode=[0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2], shadow=True)

# Add title
plt.title('FIFA World Cup Wins by Country')

# Display the plot
plt.axis('equal')  # Equal aspect ratio ensures that the pie chart is circular.
plt.show()
```

OUTPUT



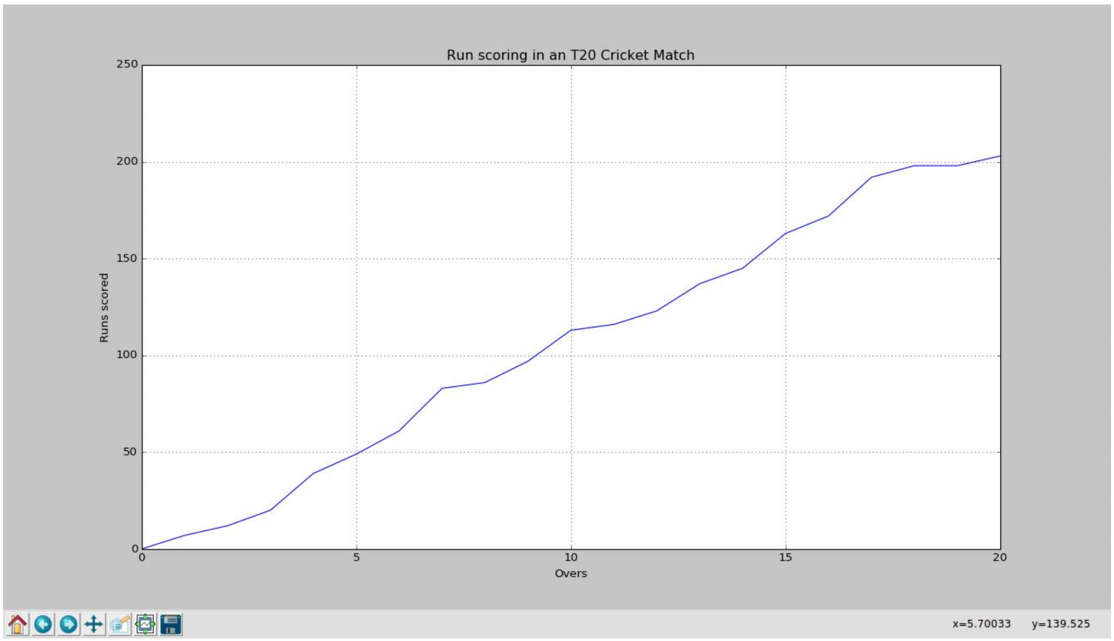
PROGRAM 6

A)Write a Python program to illustrate Linear Plotting using Matplotlib.

SOURCE CODE

```
import matplotlib.pyplot as plt

# Hypothetical data: Run rate in an T20 cricket match
overs = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
runs_scored = [0,7,12,20,39,49,61,83,86,97,113,116,123,137,145,163,172,192,198,198,203]

# Create a linear plot
plt.plot(overs, runs_scored)

# Add labels and title
plt.xlabel('Overs')
plt.ylabel('Runs scored')
plt.title('Run scoring in an T20 Cricket Match')

# Display the plot
plt.grid(True)
plt.show()
```

OUTPUT

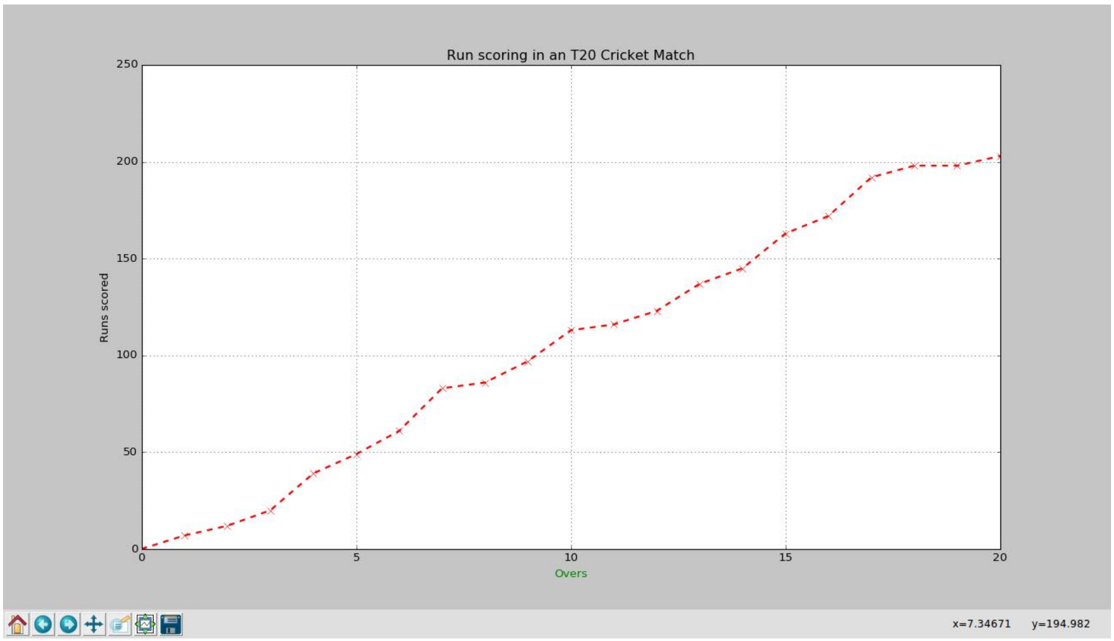


B)Write a Python program to illustrate liner plotting with line formatting using Matplotlib.

SOURCE CODE

```
import matplotlib.pyplot as plt

# Hypothetical data: Run rate in an T20 cricket match
overs = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
runs_scored = [0,7,12,20,39,49,61,83,86,97,113,116,123,137,145,163,172,192,198,198,203]

# Create a linear plot
plt.plot(overs, runs_scored, marker='X', linestyle='dashed',color='red', linewidth=2,
markerfacecolor='blue', markersize=8)

# Add labels and title
plt.xlabel('Overs', color = 'green')
plt.ylabel('Runs scored')
plt.title('Run scoring in an T20 Cricket Match')

# Display the plot
plt.grid(True)
plt.show()
```

OUTPUT



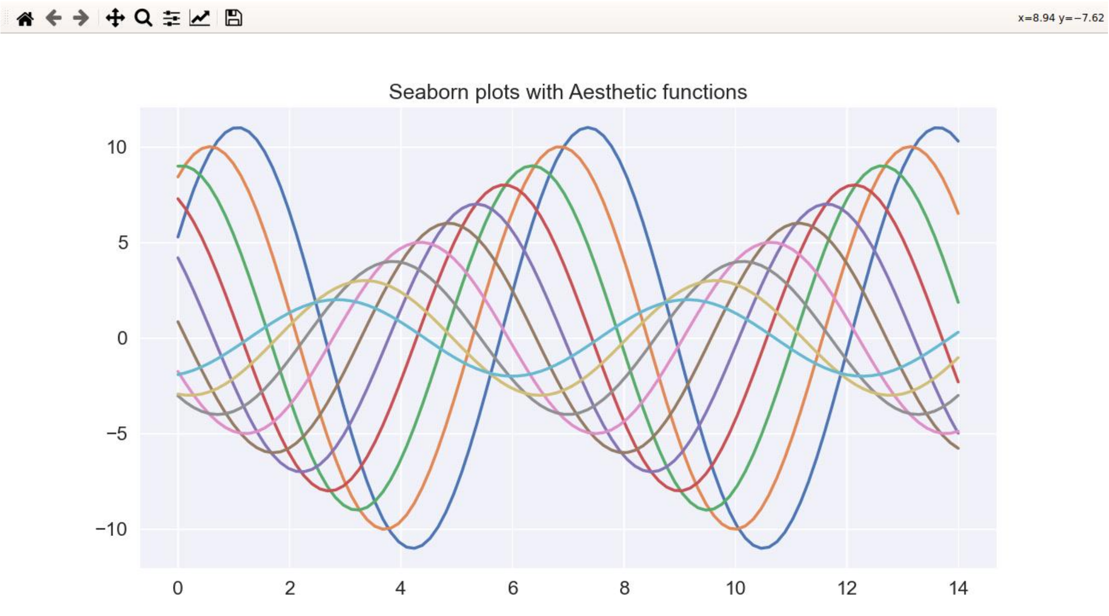
PROGRAM 7

A) Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions.

SOURCE CODE

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def sinplot(n=10):
    x = np.linspace(0, 14, 100)
    for i in range(1, n + 1):
        plt.plot(x, np.sin(x + i * .5) * (n + 2 - i))


sns.set_theme()
#sns.set_context("talk")
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth": 2.5})

sinplot()
plt.title('Seaborn plots with Aesthetic functions')
plt.show()
```

OUTPUT

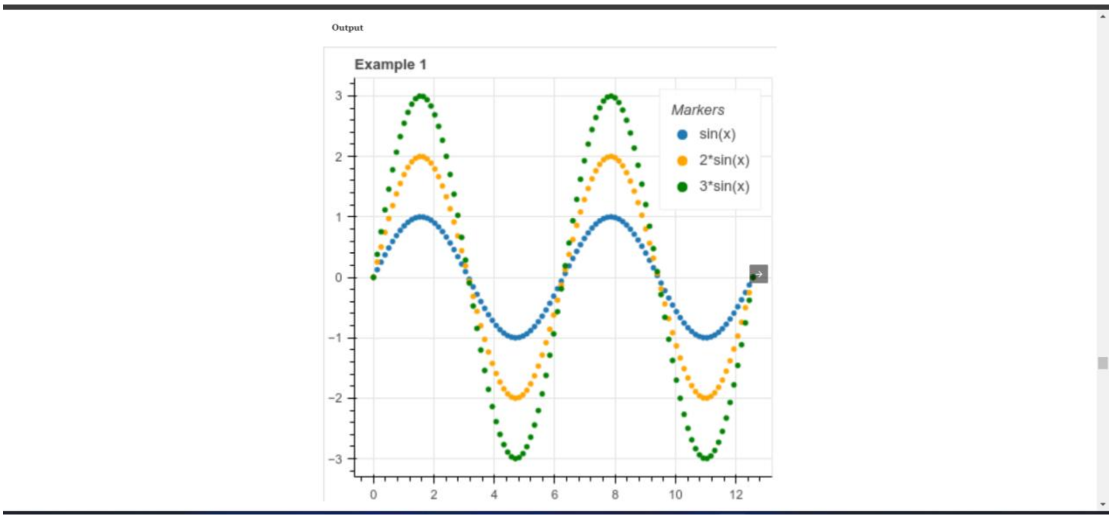


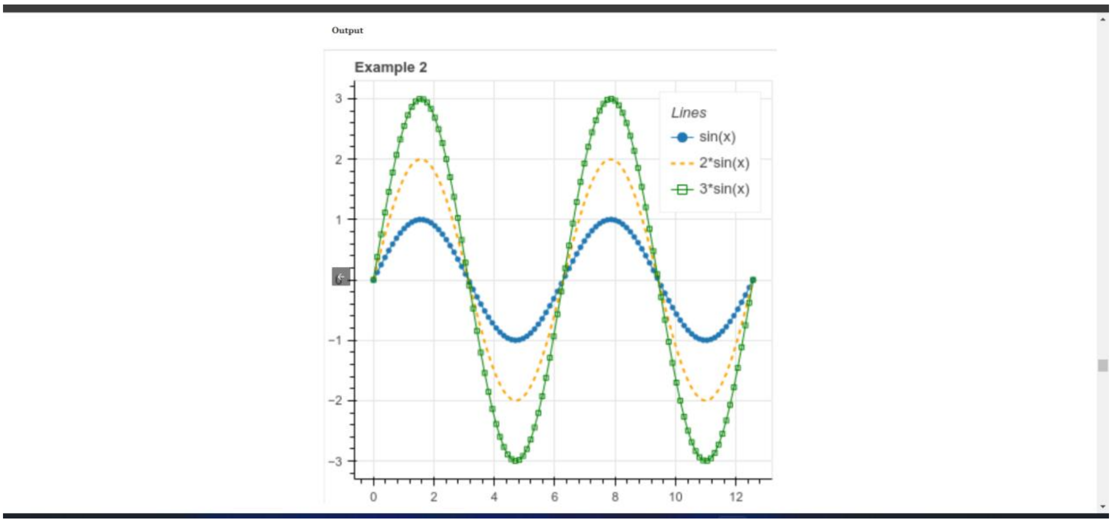
PROGRAM 8

Write a Python program to explain working with bokeh line graph using Annotations and Legends.
a)Write a Python program for plotting different types of plots using Bokeh.

```
import numpy as np

from bokeh.layouts import gridplot
from bokeh.plotting import figure, show

x = np.linspace(0, 4*np.pi, 100)
y = np.sin(x)

TOOLS = "pan,wheel_zoom,box_zoom,reset,save,box_select"

p1 = figure(title="Example 1", tools=TOOLS)

p1.circle(x,   y, legend_label="sin(x)")
p1.circle(x, 2*y, legend_label="2*sin(x)", color="orange")
p1.circle(x, 3*y, legend_label="3*sin(x)", color="green")

p1.legend.title = 'Markers'

p2 = figure(title="Example 2", tools=TOOLS)

p2.circle(x, y, legend_label="sin(x)")
p2.line(x, y, legend_label="sin(x)")

p2.line(x, 2*y, legend_label="2*sin(x)",
        line_dash=(4, 4), line_color="orange", line_width=2)

p2.square(x, 3*y, legend_label="3*sin(x)", fill_color=None, line_color="green")
p2.line(x, 3*y, legend_label="3*sin(x)", line_color="green")

p2.legend.title = 'Lines'

show(gridplot([p1, p2], ncols=2, width=400, height=400))
```

OUTPUT


Output
Example 1
Markers
sin(x)
2*sin(x)
3*sin(x)
3
2
1
0
−1
−2
−3
0
2
4
6
8
10
12

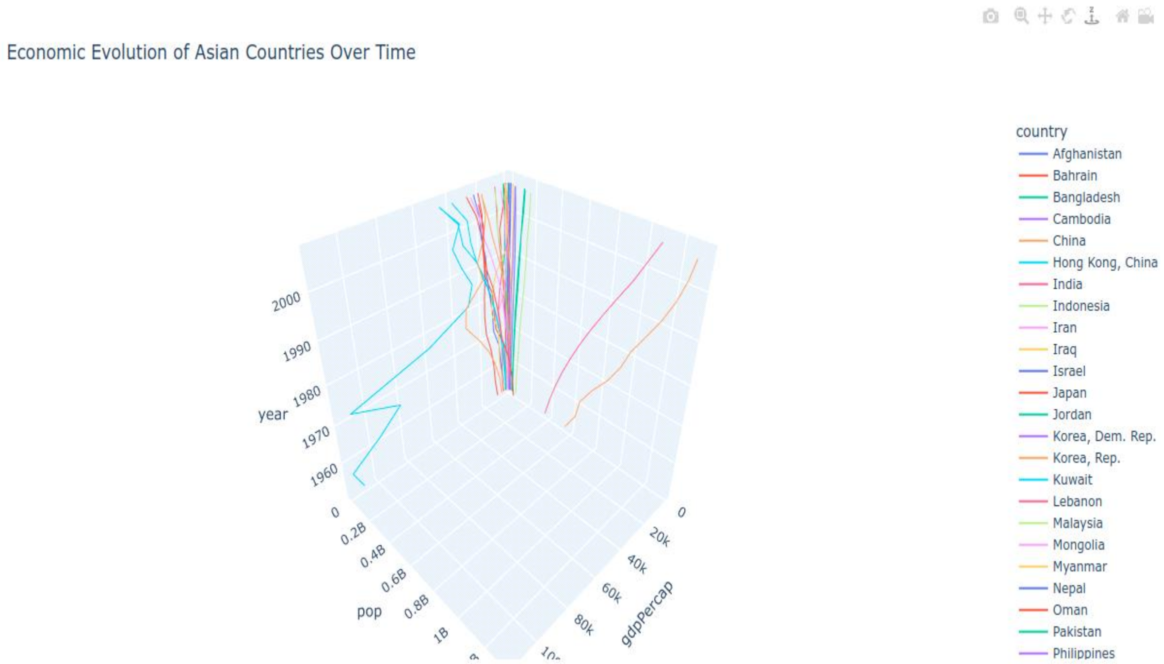

Output
Example 2
Lines
sin(x)
2*sin(x)
3*sin(x)
3
2
1
−1
−2
−3
0
2
4
6
8
10
12

PROGRAM 9

Write a Python program to draw 3D Plots using Plotly Libraries.

```
import plotly.express as px
df = px.data.gapminder().query("continent=='Asia'")
fig = px.line_3d(df, x="gdpPercap", y="pop", z="year", color='country', title='Economic
Evolution of Asian Countries Over Time')
fig.show()
```

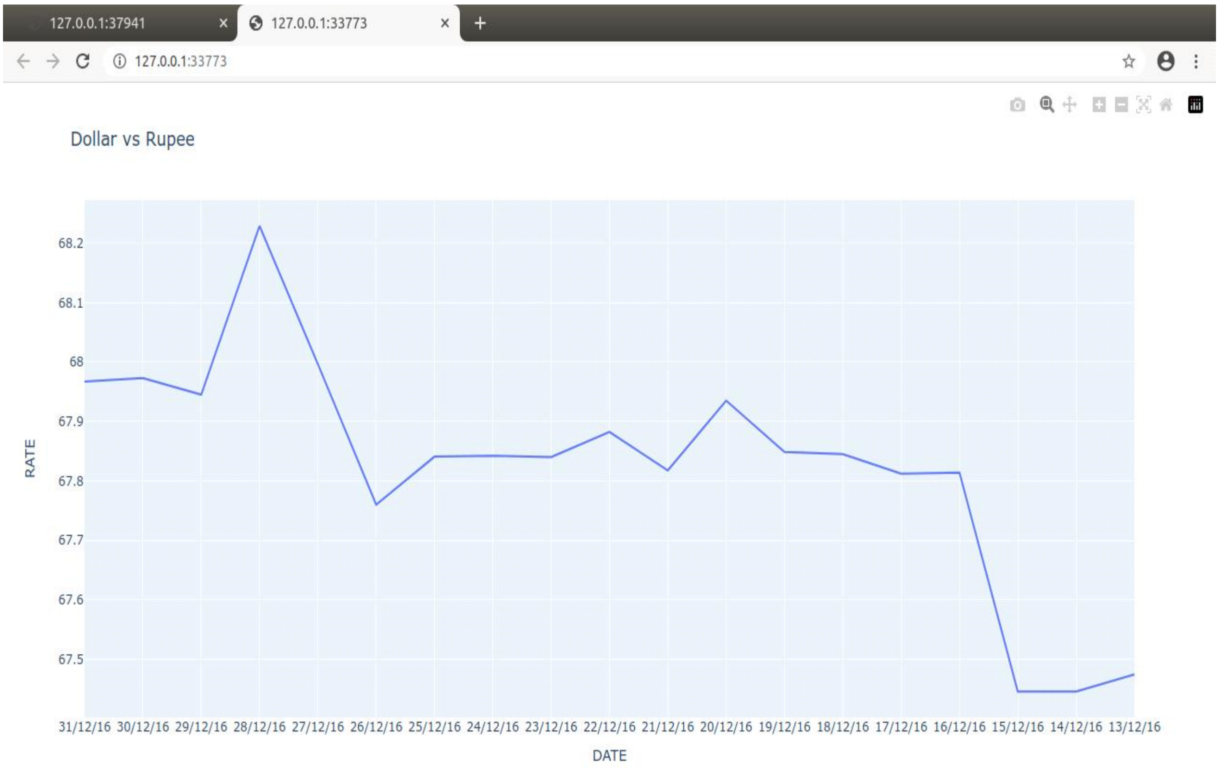
OUTPUT



PROGRAM 10

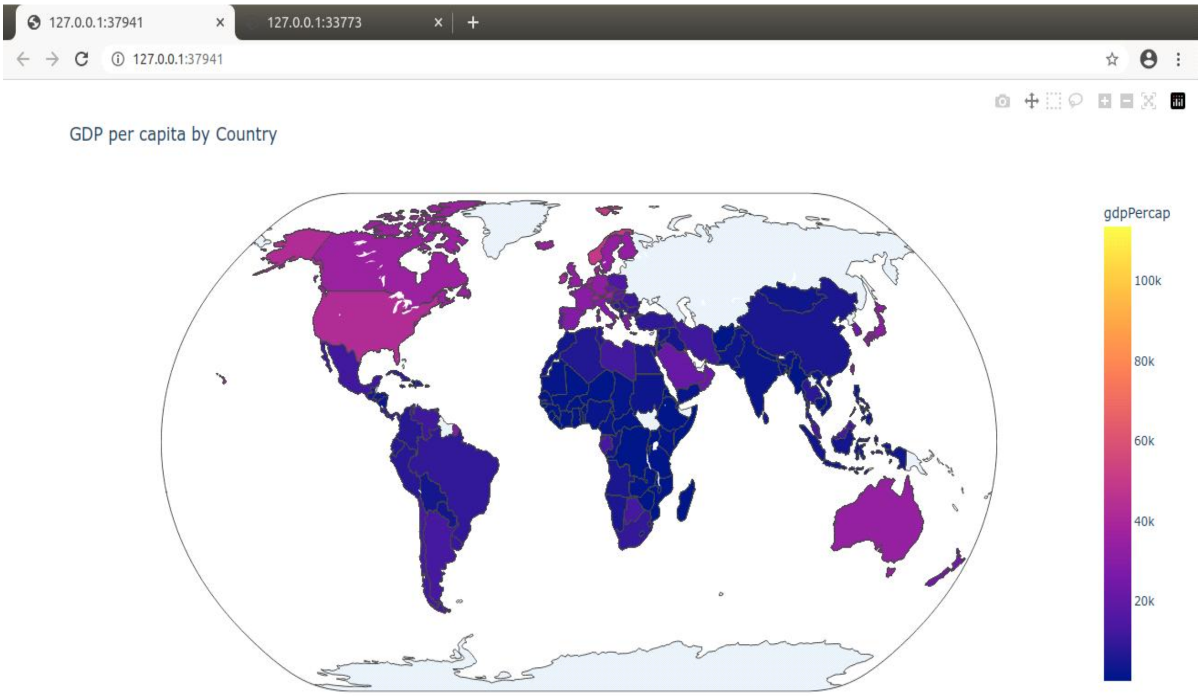
A)Write a Python program to draw Time Series using Plotly Libraries.

```
import pandas as pd
import plotly.express as px

dollar_conv = pd.read_csv('CUR_DLR_INR.csv')

fig = px.line(dollar_conv, x='DATE', y='RATE', title='Dollar vs Rupee')
fig.show()
```



B)Write a Python program for creating Maps using Plotly Libraries.

```
import plotly.express as px
import pandas as pd


# Import data from GitHub
data =
pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/gapminder_with_codes.cs
v')


# Create basic choropleth map
fig = px.choropleth(data, locations='iso_alpha', color='gdpPercap', hover_name='country',
                projection='natural earth', title='GDP per Capita by Country')
fig.show()
```


127.0.0.1:37941
127.0.0.1:33773
127.0.0.1:37941
GDP per capita by Country
gdpPercap
100k
80k
60k
40k
20k