

In [1]:

```
1 '''
2 1a) Write a python program to find the best of two test average marks out of three
3     test's marks accepted from the user.
4 '''
5 m1 = int(input("Enter marks for test1 : "))
6 m2 = int(input("Enter marks for test2 : "))
7 m3 = int(input("Enter marks for test3 : "))
8 best_of_two = sorted([m1, m2, m3], reverse=True)[:2]
9 average_best_of_two = sum(best_of_two)/2
10 print("Average of best two test marks out of three test's marks is",
11       average_best_of_two)
```

Enter marks for test1 : 40

Enter marks for test2 : 45

Enter marks for test3 : 46

Average of best two test marks out of three test's marks is 45.5

In [5]:

```
1 '''
2 1b) Develop a Python program to check whether a given number
3     is palindrome or not and also count the number of occurrences
4     of each digit in the input number.
5 '''
6 from collections import Counter
7 num = int(input("Enter a value : "))
8 #typecasting is done to convert integer value to a string
9 value=str(num)
10 if value == value[::-1]:
11     print(f'{value} is a Palindrome')
12 else:
13     print(f'{value} is Not a Palindrome')
14 counted_dict = Counter(value)
15 for key in counted_dict:
16     print(f'{key} appears {counted_dict[key]} times')
```

Enter a value : 12334

12334 is Not a Palindrome

1 appears 1 times

2 appears 1 times

3 appears 2 times

4 appears 1 times

In [13]:

```
1 '''
2 2a) Defined as a function F as  $F_n = F_{n-1} + F_{n-2}$ .
3 Write a Python program which accepts a value for N (where  $N > 0$ ) as input
4 and pass this value to the function. Display suitable error message
5 if the condition for input value is not followed.
6 '''
7 def fn(n):
8     if n <= 2:
9         return 1
10    else:
11        return fn(n-1) + fn(n-2)
12 try:
13     num = int(input("Enter a number : "))
14     if num > 0:
15         print(f' fn({num}) = {fn(num)}')
16     else:
17         print("Input should be greater than 0")
18 except ValueError:
19     print("Try with numeric value")
```

Enter a number : 6

fn(6) = 8

In [14]:

```
1 '''
2 2b) Develop a python program to convert binary to decimal,
3 octal to hexadecimal using functions.
4 '''
5 def bin2Dec(val):
6     return int(val, 2)
7 def oct2Hex(val):
8     # step1 octal to decimal
9     bin = int(val, 8)
10    # step2 decimal to hexadecimal
11    hex_dec = hex(bin)
12    return hex_dec
13 try:
14     num1 = input("Enter a binary number : ")
15     print(bin2Dec(num1))
16 except ValueError:
17     print("Invalid literal in input with base 2")
18 try:
19     num2 = input("Enter a octal number : ")
20     print(oct2Hex(num2))
21 except ValueError:
22     print("Invalid literal in input with base 8")
```

Enter a binary number : 00110

6

Enter a octal number : 56

0x2e

In [2]:

```
1 '''
2 3a) Write a Python program that accepts a sentence and
3     find the number of words, digits, uppercase letters and
4     lowercase letters.
5 '''
6 import string
7 sentence = input("Enter a sentence : ")
8 wordList = sentence.strip().split(" ")
9 print(f'This sentence has {len(wordList)} words')
10 digit_count = uppercase_count = lowercase_count = 0
11 for character in sentence:
12     if character in string.digits:
13         digit_count += 1
14     elif character in string.ascii_uppercase:
15         uppercase_count += 1
16     elif character in string.ascii_lowercase:
17         lowercase_count += 1
18 print(f'This sentence has {digit_count} digits,',
19       f'{uppercase_count} upper case letters,',
20       f'{lowercase_count} lower case letters')
```

Enter a sentence : Hello Welcome to Data Visualization with Python Lab Batch1 and Batch2

This sentence has 11 words

This sentence has 2 digits, 8 upper case letters, 49 lower case letters

In [5]:

```
1 '''
2 3b) Write a Python program to find the string similarity between two given strings
3 '''
4 from difflib import SequenceMatcher
5 str1 = input("Enter String 1 : ")
6 str2 = input("Enter String 2 : ")
7 sim = SequenceMatcher(None, str1, str2).ratio()
8 print("Similarity between strings1", str1,'and strings2',str2,'is : ',sim)
```

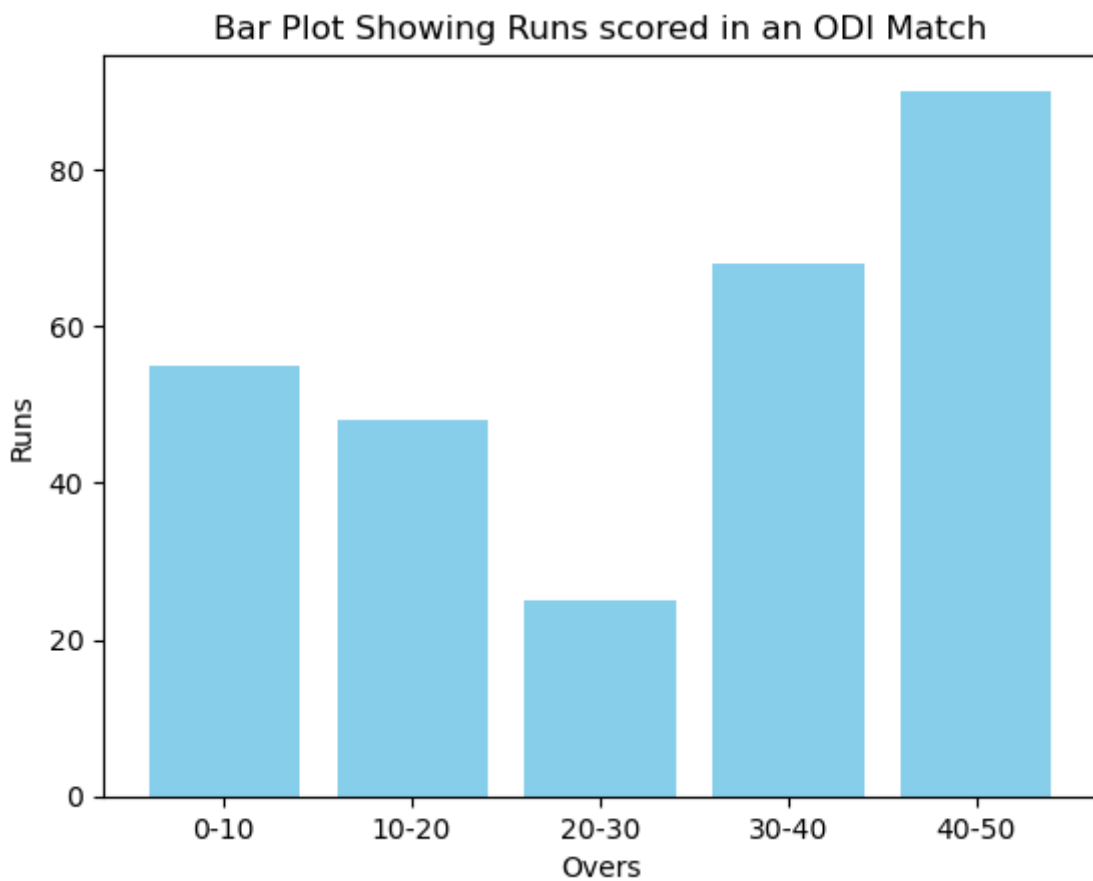
Enter String 1 : Python Exercises

Enter String 2 : Python Exercise

Similarity between strings1 Python Exercises and strings2 Python Exercise
is : 0.967741935483871

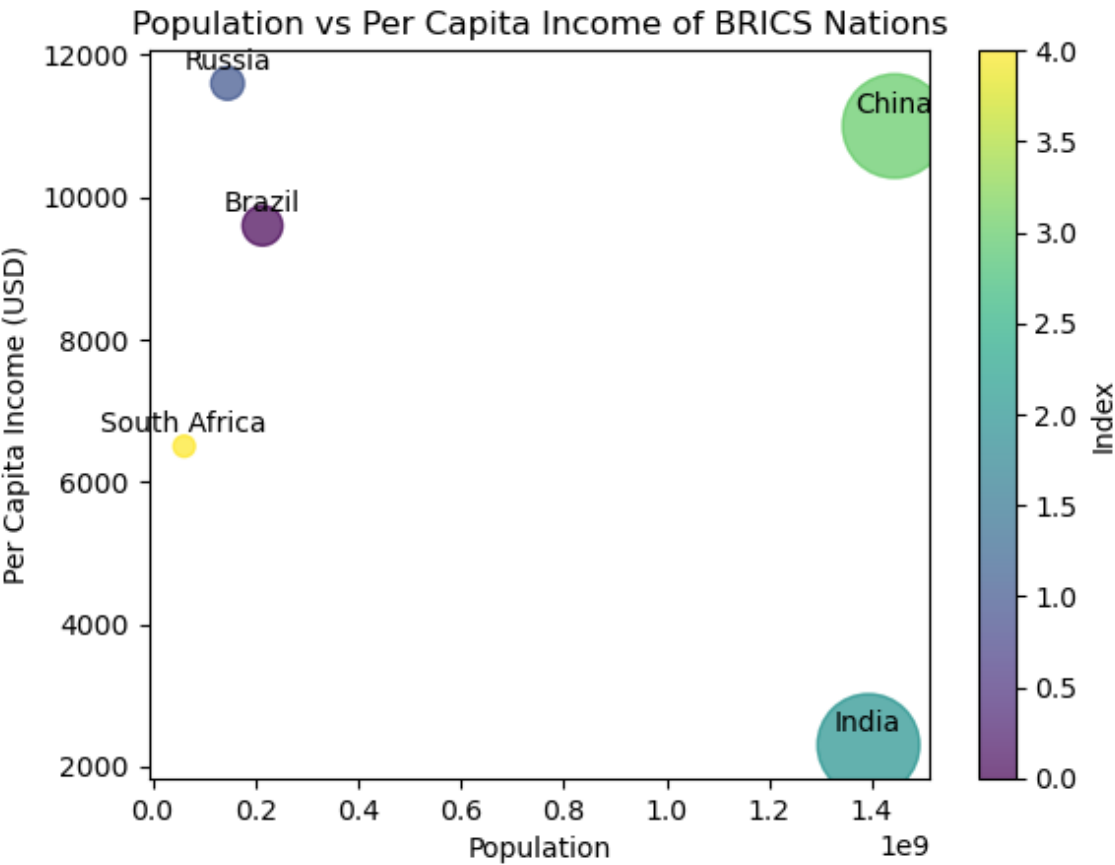
In [20]:

```
1 '''  
2 4a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.  
3 '''  
4 import matplotlib.pyplot as plt  
5 # Sample data for demonstration  
6 Odi_Overs = ['0-10', '10-20', '20-30', '30-40', '40-50']  
7 runs_scored = [55, 48, 25, 68, 90]  
8  
9 # Create a bar plot  
10 plt.bar(Odi_Overs, runs_scored, color='skyblue')  
11  
12 # Add Labels and title  
13 plt.xlabel('Overs')  
14 plt.ylabel('Runs')  
15 plt.title('Bar Plot Showing Runs scored in an ODI Match')  
16  
17 # Display the plot  
18 plt.show()
```



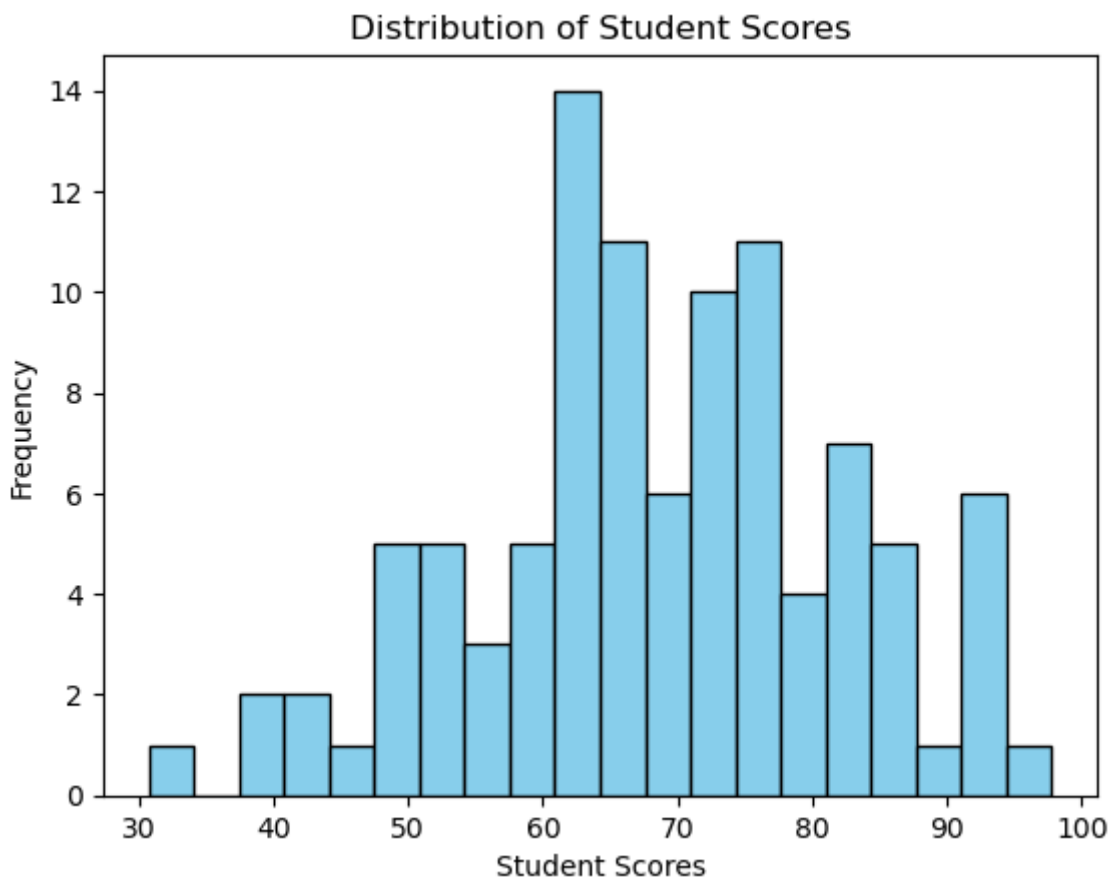
In [48]:

```
1  '''
2  4b) Write a Python program to Demonstrate how to Draw a Scatter Plot
3  using Matplotlib.
4  '''
5  import matplotlib.pyplot as plt
6  import numpy as np
7
8  # BRICS nations data (hypothetical)
9  countries = ['Brazil', 'Russia', 'India', 'China', 'South Africa']
10 population = [213993437, 145912025, 1393409038, 1444216107, 61608912]
11 per_capita_income = [9600, 11600, 2300, 11000, 6500]
12
13 # Scale the population for circle size, Scaling down for better visualization
14 circle_size = [pop / 1000000 for pop in population]
15
16 # Assign different colors based on index
17 colors = np.arange(len(countries))
18
19 # Create a scatter plot with varying circle sizes and colors
20 scatter = plt.scatter(population, per_capita_income, s=circle_size,
21                       c=colors, cmap='viridis', alpha=0.7, label='BRICS Nations')
22
23 # Annotate each point with the country name
24 for i, country in enumerate(countries):
25     plt.annotate(country, (population[i], per_capita_income[i]),
26                   textcoords="offset points", xytext=(0,5), ha='center')
27 # Add colorbar
28 plt.colorbar(scatter, label='Index')
29
30 # Add Labels and title
31 plt.xlabel('Population')
32 plt.ylabel('Per Capita Income (USD)')
33 plt.title('Population vs Per Capita Income of BRICS Nations')
34
35 # Display the plot
36 plt.show()
```



In [2]:

```
1 '''
2 5a) Write a Python program to Demonstrate how to Draw a Histogram Plot
3 using Matplotlib.
4 '''
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 # Generate random student scores (example data)
9 np.random.seed(42)
10 student_scores = np.random.normal(loc=70, scale=15, size=100)
11
12 # Create a histogram plot
13 plt.hist(student_scores, bins=20, color='skyblue', edgecolor='black')
14
15 # Add Labels and title
16 plt.xlabel('Student Scores')
17 plt.ylabel('Frequency')
18 plt.title('Distribution of Student Scores')
19
20 # Display the plot
21 plt.show()
```



In [3]:

```

1  '''
2  5b) Write a Python program to Demonstrate how to Draw a Pie Chart
3  using Matplotlib.
4  '''
5  import matplotlib.pyplot as plt
6  #Number of FIFA World Cup wins for different countries
7  countries = ['Brazil', 'Germany', 'Italy', 'Argentina',
8              'Uruguay', 'France', 'England', 'Spain']
9  wins = [5, 4, 4, 3, 2, 2, 1, 1] # Replace with actual data
10
11 # Colors for each country
12 colors = ['yellow', 'magenta', 'green', 'blue',
13           'lightblue', 'blue', 'red', 'cyan']
14
15 plt.pie(wins, labels=countries, autopct='%1.1f%%', colors=colors, startangle=90,
16         explode=[0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2], shadow=True)
17
18 # Add title
19 plt.title('FIFA World Cup Wins by Country')
20
21 # Display the plot
22 plt.axis('equal') # Equal aspect ratio ensures that the pie chart is circular.
23 plt.show()

```

