Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 1

Roll No. : 135

Name : Nikhil Rathod

1. WAP to print “Hello World”

In [2]:
```
print("Hello")
```

```
Hello
```

2. WAP to print your address i) using single print ii) using multiple print

In [7]:
```
print("Jamnagar Gujarat Street 5 pincode 361004")
print("Jamnagar")
print("Gujarat")
print("Street 5")
print("Pincode 361004")
print("India")
```

```
Jamnagar Gujarat Street 5 pincode 361004
Jamnagar
Gujarat
Street 5
Pincode 361004
India
```

3. WAP to print addition of 2 numbers (without input function)

In [1]:
```
num1 = 10
num2 = 20
res = num1 + num2
print(res)
```

```
30
```

4. WAP to calculate and print average of 2 numbers (without input function)

In [11]:
```
a = 50
b = 20
ans = a+b
avg = ans/2
print(ans)
print(avg)
```

```
70
35.0
```

5. WAP to add two number entered by user.

In [4]:
```
a = int(input("Enter Number 1 : "))
b = int(input("Enter Number 2 : "))
ans = a+b
print(ans)
```

```
40
```

6. WAP to calculate area of circle.

In [5]:
```
r = int(input("Enter Number : "))
ans = 3.14 * r * r
print(ans)
```

```
7850.0
```

7. Purposefully raise Indentation Error and Correct it.

In [12]:
```
a = 10
b = 10
if(a == b):
print("Equal")
else:
print("Not Equal!")

# a = 10
# b = 10
# if(a == b):
#     print("Equal")
# else:
#     print("Not Equal!")
```

```
  Cell In[12], line 4
    print("Equal")
    ^
IndentationError: expected an indented block after 'if' statement on line 3
```

8. WAP to calculate simple interest

In [7]:
```
p = int(input("Enter Number Amount : "))
r = int(input("Enter Number Interest : "))
n = int(input("Enter Number Month : "))
```

```
ans = ((p * r * n)/100)
print(ans)
```

```
210000.0
```

9. WAP Calculate Area and Circumference of Circle.

In [8]:
```
r = int(input("Enter Number Rate : "))
ans = 2 * 3.14 * r
print(ans)
```

```
31.400000000000002
```

10. WAP to print Multiplication table of given number.

In [13]:
```
num = int(input("Enter Number : "))
for i in range(1,11):
    print(num , " x " , i , " = ", (num*i))
```

```
5  x  1  =  5
5  x  2  =  10
5  x  3  =  15
5  x  4  =  20
5  x  5  =  25
5  x  6  =  30
5  x  7  =  35
5  x  8  =  40
5  x  9  =  45
5  x  10  =  50
```

11. WAP to calculate Area of Triangle. (hint: a = h*b*0.5)

In [14]:
```
h = int(input("Enter Height : "))
b = int(input("Enter Base : "))
res = (h*b*0.5)
print(res)
```

```
10.0
```

12. WAP to convert Degree to Fahrenheit and vice versa.

In [21]:
```
# Celsius to Fahrenheit
cel = float(input("Enter Number : "))
res = ((cel*(9/5))+32)
print(res)



# Fahrenheit to Celsius
fah = float(input("Enter Number : "))
res = (fah-32)*5/9
print(res)
```

```
113.0
45.0
```

13.WAP to calculate total marks and Percentage.

In [2]:
```
num = int(input("Enter Number of Subjects: "))
sum = 0
for i in range(1, num+1):
    sub = int(input("Enter Marks: "))
    sum = sum + sub
print("Total Marks:", sum)
print("Percentage:", (sum/(num*100))*100)
```

```
Total Marks: 400
Percentage: 80.0
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 2

Roll No. : 135

Name : Nikhil Rathod

1. WAP to check whether the given number is Positive or Negative.

In [2]:
```
num = int(input("Enter Number : "))
if num > 0:
    print("Number is Positive!")
else:
    print("Number is Negative!")
```

```
Number is Negative!
```

2. WAP to check whether the given number is Odd or Even.

In [10]:
```
num = int(input("Enter Number : "))
if num%2==0:
    print("Number is Even")
else:
    print("Number is Odd")
```

```
Number is Odd
```

3. WAP to find out Largest number from given two numbers using simple if and ternary operator.

In [6]:
```
num1 = int(input("Enter Number 1 : "))
num2 = int(input("Enter Number 2 : "))
if num1>num2:
    print("Number 1 is Big!")
else:
    print("Number 2 is big!")


# res = num1 if num1>num2 else num2
# print(res)
```

```
20
```

4. WAP to find out Largest number from given three numbers.

In [8]:
```
num1 = int(input("Enter Number 1 : "))
num2 = int(input("Enter Number 2 : "))
num3 = int(input("Enter Number 3 : "))

if num1>num2 and num1>num3:
    print("Number 1 is Big!")
elif num2>num3:
    print("Number 2 is Big")
else:
    print("Number 3 is Big")
```

```
Number 2 is Big
```

5. WAP to check whether the given year is Leap year or not.

[If a year can be divisible by 4 but not divisible by 100 then it is leap year but if it is divisible by 400 then it is leap year].

In [2]:
```
year = int(input("Enter Year : "))
if year%4==0 and year%100 != 0 or year%400==0:
    print("This year is Leap Year!")
else:
    print("This year is not Leap Year!")
```

```
This year is not Leap Year!
```

6. WAP to display the name of the Day according to the number given by the user.

In [3]:
```
choice = int(input("Enter Choice : "))
match choice:
    case 1:
        print("Monday")
    case 2:
        print("Tuesday")
    case 3:
        print("Wednesday")
    case 4:
        print("Thursday")
    case 5:
        print("Friday")
    case 6:
        print("Saturday")
    case 7:
        print("Sunday")
    case _:
        print("Invalid")
```

```
Tuesday
```

7. WAP to implement simple Calculator which performs (add,sub,mul,div) of two numbers based on user input.

In [28]:
```
num1 = int(input("Enter Number 1 : "))
num2 = int(input("Enter Number 2 : "))
choice = int(input("Enter Choice : "))
if choice == 1:
    res = num1 + num2
    print("Addition is : ",res)
elif choice == 2:
    res = num1 - num2
    print("Substraction is : ",res)
elif choice == 3:
    res = num1 * num2
    print("Multiplication is : ",res)
elif choice == 4:
    res = num1 / num2
    print("Divison is : ",res)
else:
    print("Invalid! =====  Choice Between 1 to 4")
```

```
Addition is :  110
```

8. WAP to calculate electricity bill based on following criteria. Which takes the unit from the user.

-First 1 to 50 units – Rs. 2.60/unit

-Next 50 to 100 units – Rs. 3.25/unit

-Next 100 to 200 units – Rs. 5.26/unit

-200 units – Rs. 8.45/unit

In [4]:
```
num = int(input("Enter Number : "))
if num>1 and num<50:
    print("Electricity Bill : ",(num*2.60))
elif num>=50 and num<100:
    print("Electricity Bill : ",(50*2.60)+(num-50)*3.25)
elif num>=100 and num<200:
    print("Electricity Bill : ",(50*2.60)+(50*3.25)+(num-100)*5.26)
elif num>=200:
    print("Electricity Bill : ",(50*2.60)+(50*3.25)+(100*5.26)+((num-200)*8.45))
else:
    print("Invalid!")
```

```
Electricity Bill :  555.5
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 3

Roll No. : 135

Name : Nikhil Rathod

1. WAP to print 1 to 10.

In [1]:
```
for i in range(1,11):
    print(i,end=" ")
```

```
1 2 3 4 5 6 7 8 9 10
```

2. WAP to print 1 to n.

In [2]:
```
n = int(input("Enter Number : "))
for i in range(1,n+1):
    print(i , end=" ")
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

3.WAP to print odd numbers between 1 to n.

In [3]:
```
n = int(input("Enter Number : "))
for i in range(1,n+1):
    if i%2!=0:
        print(i,end=" ")
```

```
1 3 5 7 9 11 13 15 17 19
```

4. WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 3.

In [13]:
```
n1 = int(input("Enter Number 1 :"))
n2 = int(input("Enter Number 2 : "))
for i in range(n1,n2):
    if i%2==0 and i%3!=0:
        print(i,end=" ")
```

```
10
14
16
20
22
26
28
32
34
38
40
44
46
```

5. WAP to print sum of 1 to n numbers.

In [15]:
```
n = int(input("Enter Number : "))
sum=0
for i in range(1,n+1):
    sum+=i
print(sum)
```

```
15
```

6.WAP to print sum of series 1 + 4 + 9 + 16 + 25 + 36 + ...n.

In [18]:
```
n = int(input("Enter Number : "))
for i in range(1,n+1):
    print(i*i,end=" ")
```

```
1 4 9 16 25 36 49 64 81 100
```

7. WAP to print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 ... n.

In [6]:
```
n = int(input("Enter the value of n: "))
sum = 0

for i in range(1, n + 1):
    if i % 2 == 1:
        sum += i
    else:
        sum -= i

print("Sum: ",sum)
```

```
Sum:  23
```

8. WAP to print multiplication table of given number.

In [23]:
```
n = int(input("Enter Number : "))
for i in range(1,n+1):
    print(n," x ",i," = ", (n*i))
```

```
10  x  1  =  10
10  x  2  =  20
10  x  3  =  30
10  x  4  =  40
10  x  5  =  50
10  x  6  =  60
10  x  7  =  70
10  x  8  =  80
10  x  9  =  90
10  x  10  =  100
```

9. WAP to find factorial of the given number.

In [26]:
```
n = int(input("Enter Number : "))
fact=1
for i in range(1,n+1):
    fact=fact*i
print(fact)
```

```
120
```

10. WAP to find factors of the given number.

In [27]:
```
n = int(input("Enter Number : "))
for i in range(1,n+1):
    if n%i==0:
        print(i)
```

```
1
3
5
15
```

11. WAP to find whether the given number is prime or not.

In [45]:
```
n = int(input("Enter Number : "))
count=0
for i in range(1,n+1):
    if n%i==0:
       count+=1
if count==2:
    print("Prime!")
else:
    print("Not Prime!")
```

```
Prime!
```

12. WAP to print sum of digits of given number.

In [53]:
```
num = int(input("Enter a number: "))
sum = 0

while num > 0:
    digit = num % 10
    sum+= digit
    num = num // 10
```

```
print("Sum of digits:", sum)
```

```
Sum of digits: 3
```

13. WAP to check whether the given number is palindrome or not.

In [77]:
```
n = int(input("Enter Number : "))
temp = n
total=0
while n!=0:
    digit = n%10
    total=total*10+digit
    n=n//10
if temp==total:
    print("Number is Palidrone")
else:
    print("Number is Not Palidrone")
```

```
Number is Palidrone
```

In [75]:
```
n=7
for i in range(1,n):
    for k in range(n,i,-1):
        print(" ",end="")
    for j in range(1,i):
        print("*",end=" ")
    print()
```

```
    *
   * *
  * * *
 * * * *
* * * * *
```

In [84]:
```
n=7
for i in range(1,n):
    for k in range(n,i+1):
        print(" ",end="")
    for j in range(i,1,-1):
        print("*",end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 4

Roll No. : 135

Name : Nikhil Rathod

1. WAP to check given string is palindrome or not.

In [3]:
```
s = input("Enter String : ")
s2 = s[::-1]
if s==s2:
    print("Palidrone")
else:
    print("Not Palidrone")
```

```
Not Palidrone
```

2.WAP to reverse the words in given string.

In [20]:
```
S1 = input("Enter String : ")
x = S1.split()
r = x[::-1]
print(" ".join(r))
```

```
rathod nikhil
```

3.WAP to remove ith character from given string.

In [21]:
```
s1=input("Enter String : ")
i=3
res =s1[:i]+s1[i+1:]
print(res)
```

```
nikil
```

4. WAP to find length of String without using len function..

In [25]:
```
s1=input("Enter String : ")
res = sum(1 for i in s1)
print(res)
```

```
6
```

5. WAP to print even length word in string.

In [31]:
```
S1 = input("Enter String : ")
x = S1.split()
res = [s for s in x if len(s)%2==0]
ans = " ".join(res)
print(ans)
```

6.WAP to count numbers of vowels in given string.

In [36]:
```
s1 = input("Enter String : ")
count = 0
for i in s1:
    if i in "aeiouAEIOU":
        count+=1
print(count)
```

```
3
```

7. WAP to convert given array to string.

In [40]:
```
a = ['abc','xyz','mno','qwe']
res = ' '.join(a)
print(res)
```

```
abc xyz mno qwe
```

8. Check if the password and confirm password is same or not.¶

In case of only case's mistake, show the error message.

In [39]:
```
s1 = input("Enter Password : ")
s2 = input("Enter Confirm Password : ")
if s1.casefold()==s2.casefold():
    print("Password Matched")
else:
    print("Password not Matched")
```

```
Password not Matched
```

In [ ]:
```
s1 = "@kjbflknklad_sjdk!hja&jbjkbsa"
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 5

Roll No. : 135

Name : Nikhil Rathod

1. WAP to find sum of all the elements in a List.

In [1]:
```
l1 = [1,2,3,4,5]
res = 0
for i in l1:
   res+=i
print(res)
```

```
15
```

2. WAP to find largest element in a List.

In [17]:
```
l1 = [1,2,3,4,5]
max = 0
for i in l1:
   if i>max:
    max=i
print(max)
```

```
5
```

3. WAP to interchange first and last elements in a list.

In [19]:
```
l1 = [10,20,30,40,50]
first = l1[0]
last=l1[-1]
l1[0] = last
l1[-1] = first
l1
```

Out[19]:
```
[50, 20, 30, 40, 10]
```

4. WAP to reverse the list entered by user.

In [27]:
```
l1 = [10,20,30,40,50]
for i in l1[::-1]:
    print(i,end=" ")
```

```
50 40 30 20 10
```

5. WAP to print even numbers in a list.

In [26]:
```
l5 = [10,11,12,13,14,15,16]
for i in l5:
    if i%2==0:
        print(i,end=" ")
```

```
10 12 14 16
```

6. WAP to count occurrences of an element in a list.

In [35]:
```
l6 = [10,20,30,40,20,20,20,20,30,40,50,60,70]
k = int(input("Enter Number to Find : "))
count = 0
for i in l6:
    if k==i:
        count+=1
print(k,"is in List :",count)
```

```
20 is in List : 5
```

7. WAP to extract elements with frequency greater than K.

In [39]:
```
l6 = [10,1a,1,1,1,1,1,20,30,40,20,20,20,20,30,40,50,60,70]
k = int(input("Enter Number to Find : "))
l=[]
for i in l6:
    res = l6.count(i)
    if res>k and i not in l:
        l.append(i)
print(l)
```

```
[1, 20]
```

In [49]:
```
l1 = [10,2,30,40,50,70,55]
max = 0
secondmax=0
for i in l1:
   if i>max:
    secondmax=max
    max=i
for j in l1:
   if j > secondmax or j < max:
    secondmax=i
print(secondmax)
```

```
55
```

In [ ]:
```

```

Darshan UNIVERSITY

Python for Data Science - 2305CS303

Lab - 5 Part-2

Roll No. : 135

Name : Nikhil Rathod

1. WAP to create a list of squared numbers from 0 to 9 with and without using List Comprehension.

In [44]:
```
l1 =[]
for i in range(9):
    l1.append(i*i)
l1
```

Out[44]: [0, 1, 4, 9, 16, 25, 36, 49, 64]

In [43]:
```
l1 = [i*i for i in range (9)]
l1
```

Out[43]: [0, 1, 4, 9, 16, 25, 36, 49, 64]

2. WAP to find Maximum and Minimum K elements in a given tuple.

In [29]:
```
t1 = (1,5,67,95,45,78,10)
t2=sorted(t1)
k = int(input("Enter Number : "))
print(t2[0:k])
print(t2[-k:])
```

```
[1, 5]
[78, 95]
```

3. WAP to find tuples which have all elements divisible by K from a list of tuples.

In [45]:
```
l3 = [(2,4,6),(8,12,16),(19,13,15)]
k = int(input("Enter Number : "))
flag = 1
for i in l3:
```

```
    for j in i:
        if j%k==0:
            flag=1
        else:
            flag=0
            break
    if flag==1:
        print(i)
```

```
(2, 4, 6)
(8, 12, 16)
```

4. WAP to create a list of tuples from given list having number and its cube in each tuple.

In [41]:
```
l1 = [1, 2, 3, 4, 5]
l2 = [(num, num**3) for num in l1]
print(l2)
```

```
[(1, 1), (2, 8), (3, 27), (4, 64), (5, 125)]
```

5. WAP to remove tuples of length K.

In [3]:
```
l3 = [(2,4,6),(8,12,16),(19,13)]
k = 2
ans = [i for i in l3 if len(i) != k]
ans
```

Out[3]:
```
[(2, 4, 6), (8, 12, 16)]
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 6

Roll No. : 135

Name : Nikhil Rathod

1. WAP to iterate over a set.

In [16]:
```
s2={10,70,80,20,40}
for i in s2:
    print(i)
```

```
80
20
70
40
10
```

2. WAP to convert set into list, string and tuple.

In [27]:
```
s2 = {10,20,30,40,50}
l1 = list(s2)
l1
```

Out[27]: [50, 20, 40, 10, 30]

In [28]:
```
s3 = {'Helloooo','good'}
l2 = str(s3)
l2
```

Out[28]: "{'Helloooo', 'good'}"

In [29]:
```
s4 = {10,20,30,40,50}
l3 = tuple(s4)
l3
```

Out[29]: (50, 20, 40, 10, 30)

3. WAP to check if two lists have at-least one element

In [1]:
```
s1 = {1,2,3,4,5,6}
s2 = {5,6,7,8,9,10}
s3 = s1.intersection(s2)
s3
```

Out[1]: {5, 6}

4. WAP to remove duplicates from list.

In [3]:
```
s1 = [1,2,3,4,4,5,5,6,7]
s2=[]

for i in s1:
    if i not in s2:
        s2.append(i)
s2
```

Out[3]: [1, 2, 3, 4, 5, 6, 7]

5. WAP to find unique words in the given string.

In [3]:
```
i = "abc xyz pqr stu vwx yza"
w = i.split()
for x in w:
    print(x)
```

```
abc
xyz
pqr
stu
vwx
yza
```

6. WAP to iterate over a dictionary.

In [8]:
```
d1 = {
    "name": "Nikhil",
    "age": 21,
    "city": "jamnagar"
}

for i, j in d1.items():
    print(i, ":", j)
```

```
name : Nikhil
age : 21
city : jamnagar
```

7. WAP to find the sum of all items (values) in a dictionary given by user. (Assume: values are numeric).

In [2]:
```
d1 = {}
n = int(input("Enter Number : "))
for i in range(n):
```

```
    v = int(input("Enter value : "))
    d1[k] = v
total = sum(d1.values())
print(d1)
total
```

```
{'1': 10, '2': 20}
```

Out[2]: 30

8. WAP to sort dictionary by key or value.

In [9]:
```
d1 = {}
n = int(input("Enter Number : "))
for i in range(n):
    k = input("Enter key : ")
    v = int(input("Enter value : "))
    d1[k] = v
res = dict(sorted(d1.items()))
print(res)
```

```
{'1': 10, '2': 20, '3': 30}
```

9. WAP to handle missing keys in dictionaries.

- Example : Given, dict1 = {'a': 5, 'c': 8, 'e': 2}

- if you look for key = 'd', the message given should be 'Key Not Found', otherwise print the value of 'd' in dict1.

In [23]:
```
# d1 = {}
d1 = {'a': 5, 'c': 8, 'e': 2}
# n = int(input("Enter Number : "))
# for i in range(n):
#     k = input("Enter key : ")
#     v = int(input("Enter value : "))
#     d1[k] = v
k = input("Enter key : ")
if k not in d1:
    print("Key Not Found")
else:
    d1[k]
```

In [ ]:
```

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 7

Roll No. : 135

Name : Nikhil Rathod

1. WAP to count simple interest using function.

In [2]:
```
def simple_interest(p,r,n):
    return (p*r*n)/100
simple_interest(5000,2,3)
```

Out[2]: 300.0

2. Write a function to calculate BMI given mass and height. (BMI = mass/h**2)

In [3]:
```
def BMI(m,h):
    return m/(h**2)
BMI(200,5)
```

Out[3]: 8.0

3. WAP that defines a function to add first n numbers.

In [22]:
```
def sumofn(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total
sumofn(5)
```

Out[22]: 15

4. WAP to find maximum number from given two numbers using function.

In [6]:
```
def maxnum(num1,num2):
    if num1>num2:
```

```
        return num1
    else:
        return num2
maxnum(10,20)
```

Out[6]: 20

5. Write a function that returns True if the given string is Palindrome or False otherwise.

In [10]:
```
def palidrone(s):
    if s==s[::-1]:
        return "String is Palindrone"
    else:
        return "String is NOT Palindrone"
palidrone("nayan")
```

Out[10]: 'String is Palindrone'

6. Write a function that returns the sum of all the elements of the list.

In [1]:
```
def sumoflist(numbers):
    total = 0
    for num in numbers:
        total += num
    return total
l = [10, 20, 30, 40]
print("Sum of list:", sumoflist(l))
```

```
Sum of list: 100
```

7. WAP that defines a function which returns 1 if the number is prime otherwise return 0.

In [24]:
```
def isPrime(n):
    if n <= 1:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True
isPrime(15)
```

Out[24]: False

8. Write a function that returns the list of Prime numbers between given two numbers.

In [32]:
```
l1 = []
def PrimeRange(m,n):
    for i in range(m,n+1):
        if (isPrime(i)):
            l1.append(i)
```

```
    print(l1)
PrimeRange(2,50)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

9. WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...).

In [46]:
```
def fibo(n):
    for i in range(n):
        a = 0
        b = 1
        print(a,end=" ")
        n = a + b
        a = b
        b = n
fibo(8)
```

```
0 0 0 0 0 0 0 0
```

10. WAP to find the factorial of a given number using recursion.

In [2]:
```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

num = int(input("Enter a number: "))

if num < 0:
    print("Not Allowed")
else:
    print(f"The factorial of {num} is {factorial(num)}")
```

```
The factorial of 5 is 120
```

11. WAP to implement simple calculator using lamda function.

In [4]:
```
add = lambda a, b: a + b
sub = lambda a, b: a - b
mul = lambda a, b: a * b
div = lambda a, b: a / b if b != 0 else "Division by zero not allowed"
choice = input("Enter operator (+, -, *, /): ")
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

if choice == '+':
    print("Result:", add(num1, num2))
elif choice == '-':
    print("Result:", sub(num1, num2))
elif choice == '*':
    print("Result:", mul(num1, num2))
elif choice == '/':
    print("Result:", div(num1, num2))
```

```
else:
    print("Invalid operator")
```

```
Result: 30.0
```

In [ ]:

Python Programming - 2301CS404

Lab - 7 (Part-2)

User Defined Function

12. Write a function to calculate the sum of the first element of each tuples inside the list.

In [24]:
```
t1 = [(10, 456), (20, 100), (30, 678)]
def sumoffirstelement(t1):
    sum=0
    for i in t1:
        sum += i[0]
    return sum
sumoffirstelement(t1)
```

Out[24]: 60

13. Write a function to get the name of the student based on the given rollno.

Example: Given dict1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'} find name of student whose rollno = 103

In [12]:
```
s1 = {101:'Ajay', 102:'Rahul', 103:'Jay', 104:'Pooja'}
k = 103
def studentname(s1, k):
    return s1.get(k)
print(studentname(s1, k))
```

```
Jay
```

14. Write a function to get the sum of the scores ending with zero.

Example : scores = [200, 456, 300, 100, 234, 678]

Ans = 200 + 300 + 100 = 600

In [22]:
```
l2 = [200, 456, 300, 100, 234, 678]
def sumscore(l2):
    total = 0
    for i in l2:
        if i%10==0:
            total += i
    return total
n = sumscore(l2)
print(n)
```

```
600
```

15. Write a function to invert a given Dictionary.

hint: keys to values & values to keys

Before : {'a': 10, 'b':20, 'c':30, 'd':40}

After : {10:'a', 20:'b', 30:'c', 40:'d'}

In [33]:
```
s1 = {'a': 10, 'b':20, 'c':30, 'd':40,'a':10,'b':20}
def InvertDict(s1):
    s2 = {}
    for i,j in s1.items():
        s2[j]=i
    return s2
InvertDict(s1)

# def InvertDict():
#     i=9
#     j=10
#     temp = i
#     i = j
#     j = temp
#     return i,j
# i,j=InvertDict()
# print("i , j",i,j)
```

Out[33]:
```
{10: 'a', 20: 'b', 30: 'c', 40: 'd'}
```

16. Write a function that returns the number of uppercase and lowercase letters in the given string.

example : Input : s1 = AbcDEfgh ,Ouptput : no_upper = 3, no_lower = 5

In [35]:
```
s1="AbcDEfgh"
def ul():
    lower=0
    upper=0
    for i in s1:
        if i.islower():
            upper+=1
        else:
            lower+=1
    print("UpperCase:",upper)
```

```
    print("LowerCase:",lower)
ul()
```

```
UpperCase: 5
LowerCase: 3
```

17.Write a lambda function to get smallest number from the given two numbers.

In [40]:
```
s1 = lambda a,b : a if a < b else b
s1(32,200)
```

Out[40]: 32

18. For the given list of names of students, extract the names having more that 7 characters. Use filter().

In [42]:
```
l1 = ["abcdef","adhbaahd","qjwkdbkp","iuqwd"]
student = list(filter(lambda x: len(x) > 7, l1))
student
```

Out[42]: ['adhbaahd', 'qjwkdbkp']

19. For the given list of names of students, convert the first letter of all the names into uppercase. use map().

In [48]:
```
l1 = ["abc","xyz","pqr","klm"]
student = list(map(lambda x: x[0].upper() + x[1:], l1))
student
```

Out[48]: ['Abc', 'Xyz', 'Pqr', 'Klm']

20. Write udfs to call the functions with following types of arguments:

1. Positional Arguments
2. Keyword Arguments
3. Default Arguments
4. Variable Legngth Positional(*args) & variable length Keyword Arguments (*kwargs)
5. Keyword-Only & Positional Only Arguments

In [1]:
```
#Positional Arguments
def demo(name, age):
    print(f"Hello {name}, you are {age} years old.")

demo("Nikhil", 22)
```

```
Hello Nikhil, you are 22 years old.
```

In [2]:
```
#Keyword Arguments
def demo2(name, age):
    print(f"Hello {name}, you are {age} years old.")
demo2(age=20, name="abc")
```

```
Hello abc, you are 20 years old.
```

In [3]:
```python
#Default Arguments
def demo3(name, age=18):
    print(f"Hello {name}, you are {age} years old.")
demo3("xyz")
demo3("abc", 21)
```

```
Hello xyz, you are 18 years old.
Hello abc, you are 21 years old.
```

In [4]:
```python
#Variable Legngth Positional(*args)
def add(*args):
    total = sum(args)
    print("Total:", total)
add(10, 20, 30)
add(5, 15)
```

```
Total: 60
Total: 20
```

In [6]:
```python
# variable length Keyword Arguments (**kwargs)
def show_details(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")
show_details(name="aabc", age=25, city="jamnagar")
```

```
name: aabc
age: 25
city: jamnagar
```

In [7]:
```python
#Keyword-Only & Positional Only Arguments
def details(id, /, name, *, age):
    print(f"ID: {id}, Name: {name}, Age: {age}")
details(101, "abc", age=25)
```

```
ID: 101, Name: abc, Age: 25
```

In [ ]:
```python

```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 8

Roll No. : 135

Name : Nikhil Rathod

1. import numpy library.

In [1]:
```
import numpy as np
```

2.Create an array of 10 zeros

In [4]:
```
np.zeros(10)
```

Out[4]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

3. Create an array of 10 ones.

In [5]:
```
np.ones(10)
```

Out[5]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

4. Create an array of 10 fives

In [28]:
```
np.full(10,5)
```

Out[28]:
```
array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
```

5. Create an array of integers from 10 to 50.

In [10]:
```
np.arange(1,51,1)
```

Out[10]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50])
```

6. Create an array of all the even integers from 10 to 50.

In [15]:
```
np.arange(10,51,2)
```

Out[15]:
```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
       44, 46, 48, 50])
```

7. Create a 3x3 matrix with values ranging from 0 to 8.

In [24]:
```
a1 = np.array([[[0,1,2],[3,4,5],[6,7,8]]])
a1
```

Out[24]:
```
array([[[0, 1, 2],
        [3, 4, 5],
        [6, 7, 8]]])
```

8. Create a 3x3 identity matrix.

In [31]:
```
a2 = np.eye(3)
print(a2)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

9. Use Numpy to generate a random number between 0 and 1

In [34]:
```
a3 = np.random.randint(0,1,2)
print(a3)
```

```
[0 0]
```

10. Use Numpy to generate an array of 25 random numbers sampled from a standard normal distribution.

In [37]:
```
rand_no = np.random.randn(25)
print(rand_no)
```

```
[-0.6664252  -1.32417056  0.45843106 -2.6578759  -0.4525991   2.28058298
  1.79945796  1.21785238 -0.09991865 -0.32802129 -0.28432504  0.77023519
  0.69401549 -1.10606506 -0.20739018 -0.35200384  0.83649324  0.16638486
  0.96126215  0.79823611 -0.44698955  1.29377777  0.87299239  0.98673143
  1.30201239]
```

11. Create linspace array

In [39]:
```
np.linspace(0,10,12)
```

Out[39]:
```
array([ 0.        ,  0.90909091,  1.81818182,  2.72727273,  3.63636364,
        4.54545455,  5.45454545,  6.36363636,  7.27272727,  8.18181818,
        9.09090909, 10.        ])
```

12. Create an array of 20 linearly spaced points between 0 and 1.

In [41]:
```
a5 = np.linspace(0,1,20)
a5
```

Out[41]:
```
array([0.        , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
       0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
       0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
       0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.        ])
```

13. Create Random Integer Array

In [45]:
```
arr = np.random.randint(0,100, size = 10)
arr
```

Out[45]:
```
array([97, 12, 86, 35, 36, 54, 66, 63, 17, 79])
```

14. Create Random Integer Array and Reshape that Array

In [54]:
```
a = np.random.randint(0,100,12).reshape(3,4)
a
```

Out[54]:
```
array([[39, 80, 57, 97],
       [16, 19, 87, 57],
       [70,  2, 85, 14]])
```

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 9

Roll No. : 135

Name : Nikhil Rathod

1. Create a Pandas Series containing names of 5 students.

In [2]:
```
import pandas as pd
stu = pd.Series(['abc','xyz','pqr','tuv','lop'])
stu
```

Out[2]:
```
0    abc
1    xyz
2    pqr
3    tuv
4    lop
dtype: object
```

2. Create a Series with student roll numbers as index and their IAT scores as values..

In [3]:
```
import numpy as np
stu1 = pd.Series([89,78,61,13,23],[1,2,3,4,5])
stu1
```

Out[3]:
```
1    89
2    78
3    61
4    13
5    23
dtype: int64
```

3. Create a time series (daily) from 2025-08-01 to 2025-08-10 representing attendance tracking for a student.

In [8]:
```
date='2025-08-01'
c = pd.to_datetime(date)
temp = c + pd.to_timedelta(range(10),unit='D')
pd.Series(['A','P','P','P','A','P','A','A','P','P',],temp)
```

Out[8]:
```
2025-08-01    A
2025-08-02    P
2025-08-03    P
2025-08-04    P
2025-08-05    A
2025-08-06    P
2025-08-07    A
2025-08-08    A
2025-08-09    P
2025-08-10    P
dtype: object
```

4. Create a DataFrame for 10 students with the following columns: Roll No, Name, PDS, CA, CN, IAT.

(Use NumPy random module to generate scores)

In [42]:
```
studets = {101:"nikhil",102:"ajay",103:"harsh",104:"ronak",105:"uday"}
data = pd.DataFrame(np.random.randint(50,100,20).reshape(5,4),range(0,5),['PDS',
data["Roll"] = studets.keys()
data["Name"] = studets.values()
data
```

Out[42]:

	PDS	CA	CN	IAT	Roll	Name
0	63	54	81	73	101	nikhil
1	89	81	98	85	102	ajay
2	87	53	80	60	103	harsh
3	76	59	86	80	104	ronak
4	82	78	88	79	105	uday

5. Display the first 3 rows of the DataFrame.

In [43]:
```
data.head(3)
```

Out[43]:

	PDS	CA	CN	IAT	Roll	Name
0	63	54	81	73	101	nikhil
1	89	81	98	85	102	ajay
2	87	53	80	60	103	harsh

6. Display the last 2 rows of the DataFrame.

In [44]:
```
data.tail(2)
```

Out[44]:

	PDS	CA	CN	IAT	Roll	Name
3	76	59	86	80	104	ronak
4	82	78	88	79	105	uday

7. Use .describe() to summarize the numeric data.

In [46]:
```
data.describe()
```

Out[46]:

	PDS	CA	CN	IAT	Roll
count	5.000000	5.000000	5.000000	5.000000	5.000000
mean	79.400000	65.000000	86.600000	75.400000	103.000000
std	10.454664	13.472194	7.197222	9.607289	1.581139
min	63.000000	53.000000	80.000000	60.000000	101.000000
25%	76.000000	54.000000	81.000000	73.000000	102.000000
50%	82.000000	59.000000	86.000000	79.000000	103.000000
75%	87.000000	78.000000	88.000000	80.000000	104.000000
max	89.000000	81.000000	98.000000	85.000000	105.000000

8. Select only the Name column.

In [47]:
```
data['Name']
```

Out[47]:
```
0    nikhil
1      ajay
2     harsh
3     ronak
4      uday
Name: Name, dtype: object
```

9. Select the columns PDS, CN, and IAT.

In [49]:
```
data[['PDS','CN','IAT']]
```

Out[49]:

	PDS	CN	IAT
0	63	81	73
1	89	98	85
2	87	80	60
3	76	86	80
4	82	88	79

10. Select the row with Roll No = 105 using loc.

In [52]:
```
data.loc[4]
```

Out[52]:
```
PDS       82
CA        78
CN        88
IAT       79
Roll     105
Name    uday
Name: 4, dtype: object
```

11. Select the 4th row using iloc.

In [53]:
```
data.iloc[4]
```

Out[53]:
```
PDS       82
CA        78
CN        88
IAT       79
Roll     105
Name    uday
Name: 4, dtype: object
```

12. Select students with marks in PDS > 80.

In [55]:
```
data.loc[data['PDS']>80]
```

Out[55]:

	PDS	CA	CN	IAT	Roll	Name
1	89	81	98	85	102	ajay
2	87	53	80	60	103	harsh
4	82	78	88	79	105	uday

13. Select students with marks in CA < 70.

In [56]:
```
data.loc[data['CA']>70]
```

Out[56]:

	PDS	CA	CN	IAT	Roll	Name
1	89	81	98	85	102	ajay
4	82	78	88	79	105	uday

14. Select students with marks in CN > 85 and PDS > 80

In [57]:
```
data.loc[(data['CN']>85) & (data['PDS']>80)]
```

Out[57]:

	PDS	CA	CN	IAT	Roll	Name
1	89	81	98	85	102	ajay
4	82	78	88	79	105	uday

15. Add a new column Total Marks = PDS + CA + CN + IAT.

In [59]:
```
data['total']=data['PDS']+data['CA']+data['CN']+data['IAT']
data
```

Out[59]:

	PDS	CA	CN	IAT	Roll	Name	total
0	63	54	81	73	101	nikhil	271
1	89	81	98	85	102	ajay	353
2	87	53	80	60	103	harsh	280
3	76	59	86	80	104	ronak	301
4	82	78	88	79	105	uday	327

16. Create a new DataFrame of students with Total Marks > 320.

In [60]:
```
data.loc[data['total']>320]
```

Out[60]:

	PDS	CA	CN	IAT	Roll	Name	total
1	89	81	98	85	102	ajay	353
4	82	78	88	79	105	uday	327

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 10

Roll No. : 135

Name : Nikhil Rathod

Student Score (.csv file)

1. Load the file student_scores.csv.

In [1]:
```
import pandas as pd
df = pd.read_csv('students_score.csv')
df
```

Out[1]:

	RollNo	Name	Math	Science	English
0	101	Aman	78	85	90
1	102	Riya	65	82	75
2	103	Kiran	90	88	92
3	104	Ravi	70	79	85
4	105	Meera	88	92	91
5	106	John	81	87	93
6	107	Sara	77	90	89
7	108	Tom	69	73	80
8	109	Alice	84	88	85
9	110	Neha	72	78	76

2. Show the first 5 rows.

In [11]:
```
df.head()
```

Out[11]:

	RollNo	Name	Math	Science	English
0	101	Aman	78	85	90
1	102	Riya	65	82	75
2	103	Kiran	90	88	92
3	104	Ravi	70	79	85
4	105	Meera	88	92	91

3. Display the index and column names.

In [19]:
```
df.columns
# df.index
```

Out[19]:
```
Index(['RollNo', 'Name', 'Math', 'Science', 'English'], dtype='object')
```

4. Get descriptive statistics using .describe().

In [12]:
```
df.describe()
```

Out[12]:

	RollNo	Math	Science	English
count	10.00000	10.00000	10.000000	10.00000
mean	105.50000	77.40000	84.200000	85.60000
std	3.02765	8.40899	6.033241	6.60303
min	101.00000	65.00000	73.000000	75.00000
25%	103.25000	70.50000	79.750000	81.25000
50%	105.50000	77.50000	86.000000	87.00000
75%	107.75000	83.25000	88.000000	90.75000
max	110.00000	90.00000	92.000000	93.00000

5. Select the Name and Math columns.

In [20]:
```
df[['Name','Math']]
```

Out[20]:

	Name	Math
0	Aman	78
1	Riya	65
2	Kiran	90
3	Ravi	70
4	Meera	88
5	John	81
6	Sara	77
7	Tom	69
8	Alice	84
9	Neha	72

6. Find all students who scored more than 80 in Science.

In [26]:
```
df[df['Science']>80]
```

Out[26]:

	RollNo	Name	Math	Science	English
0	101	Aman	78	85	90
1	102	Riya	65	82	75
2	103	Kiran	90	88	92
4	105	Meera	88	92	91
5	106	John	81	87	93
6	107	Sara	77	90	89
8	109	Alice	84	88	85

7. Find all students with English < 75.

In [29]:
```
df[df['English']<75]
```

Out[29]:

	RollNo	Name	Math	Science	English

8. Extract the last 3 rows.

In [30]:
```
df.tail(3)
```

Out[30]:

	RollNo	Name	Math	Science	English
7	108	Tom	69	73	80
8	109	Alice	84	88	85
9	110	Neha	72	78	76

9. Sort the DataFrame by Math in descending order.

(Hint : use df.sort_values(by = "column_name", ascending = True/False))

In [13]:
```
df.sort_values(by='Math',ascending = False)
```

Out[13]:

	RollNo	Name	Math	Science	English
2	103	Kiran	90	88	92
4	105	Meera	88	92	91
8	109	Alice	84	88	85
5	106	John	81	87	93
0	101	Aman	78	85	90
6	107	Sara	77	90	89
9	110	Neha	72	78	76
3	104	Ravi	70	79	85
7	108	Tom	69	73	80
1	102	Riya	65	82	75

10. Set RollNo as the index and rename it "Student ID".

In [17]:
```
#df.set_index('RollNo')
#df.index.name = 'Student ID'
df
```

Out[17]:

Student ID	RollNo	Name	Math	Science	English
0	101	Aman	78	85	90
1	102	Riya	65	82	75
2	103	Kiran	90	88	92
3	104	Ravi	70	79	85
4	105	Meera	88	92	91
5	106	John	81	87	93
6	107	Sara	77	90	89
7	108	Tom	69	73	80
8	109	Alice	84	88	85
9	110	Neha	72	78	76

11. Reset the index back.

In [18]:
```
df.reset_index()
```

Out[18]:

	Student ID	RollNo	Name	Math	Science	English
0	0	101	Aman	78	85	90
1	1	102	Riya	65	82	75
2	2	103	Kiran	90	88	92
3	3	104	Ravi	70	79	85
4	4	105	Meera	88	92	91
5	5	106	John	81	87	93
6	6	107	Sara	77	90	89
7	7	108	Tom	69	73	80
8	8	109	Alice	84	88	85
9	9	110	Neha	72	78	76

12. Add a new column Total = Math + Science + English.

In [2]:
```
df['Total']=df.Math+df.Science+df.English
df
```

Out[2]:

	RollNo	Name	Math	Science	English	Total
0	101	Aman	78	85	90	253
1	102	Riya	65	82	75	222
2	103	Kiran	90	88	92	270
3	104	Ravi	70	79	85	234
4	105	Meera	88	92	91	271
5	106	John	81	87	93	261
6	107	Sara	77	90	89	256
7	108	Tom	69	73	80	222
8	109	Alice	84	88	85	257
9	110	Neha	72	78	76	226

13. Find the student with the highest Total score.

In [23]:
```
df.sort_values(by="Total", ascending=False)
```

Out[23]:

Student ID	RollNo	Name	Math	Science	English	Total
4	105	Meera	88	92	91	271
2	103	Kiran	90	88	92	270
5	106	John	81	87	93	261
8	109	Alice	84	88	85	257
6	107	Sara	77	90	89	256
0	101	Aman	78	85	90	253
3	104	Ravi	70	79	85	234
9	110	Neha	72	78	76	226
1	102	Riya	65	82	75	222
7	108	Tom	69	73	80	222

14. Get the Top 3 students with the highest total score.

In [3]:
```
df.sort_values(by="Total", ascending=False).head(3)
```

Out[3]:

	RollNo	Name	Math	Science	English	Total
4	105	Meera	88	92	91	271
2	103	Kiran	90	88	92	270
5	106	John	81	87	93	261

15. Get the average marks in each subject.

In [4]:
```
df[["Math", "Science", "English"]].mean()
```

Out[4]:
```
Math       77.4
Science    84.2
English    85.6
dtype: float64
```

In [ ]:

Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python for Data Science - 2305CS303

Lab - 11

Roll No. : 135

Name : Nikhil Rathod

GroupBy

In [2]:
```
students = {
    'RollNo': [101, 102, 103, 104, 105, 106],
    'Name': ['Aarav', 'Diya', 'Ishaan', 'Meera', 'Kabir', 'Anaya'],
    'Dept': ['CSE', 'CSE', 'ECE', 'ECE', 'ME', 'CSE'],
    'Math': [88, 92, None, 74, 69, 85],
    'Science': [91, None, 78, 84, 76, 89],
    'English': [85, 87, 80, None, 74, 90]
}
students
```

Out[2]:
```
{'RollNo': [101, 102, 103, 104, 105, 106],
 'Name': ['Aarav', 'Diya', 'Ishaan', 'Meera', 'Kabir', 'Anaya'],
 'Dept': ['CSE', 'CSE', 'ECE', 'ECE', 'ME', 'CSE'],
 'Math': [88, 92, None, 74, 69, 85],
 'Science': [91, None, 78, 84, 76, 89],
 'English': [85, 87, 80, None, 74, 90]}
```

1. Group students by Dept and find the average marks in each subject.

In [12]:
```
import pandas as pd

df = pd.DataFrame(students)
df

df1 = df.groupby('Dept')
df1[['Math','Science','English']].mean()
```

Out[12]:

Dept	Math	Science	English
CSE	88.333333	90.0	87.333333
ECE	74.000000	81.0	80.000000
ME	69.000000	76.0	74.000000

2. Find the highest Math score in each department.

In [13]:
```
df1['Math'].max()
```

Out[13]:
```
Dept
CSE    92.0
ECE    74.0
ME     69.0
Name: Math, dtype: float64
```

3. Count how many students belong to each department.

In [14]:
```
df1[['Math','Science','English']].count()
```

Out[14]:

Dept	Math	Science	English
CSE	3	2	3
ECE	1	2	1
ME	1	1	1

4. Compute the minimum, maximum, and mean of Science marks.

In [15]:
```
df1['Science'].min()
df1['Science'].max()
df1['Science'].mean()
```

Out[15]:
```
Dept
CSE    90.0
ECE    81.0
ME     76.0
Name: Science, dtype: float64
```

5. For each department, apply multiple aggregations:

Math: mean, max

Science: min, count

In [20]:
```
df1[['Math','Science','English']].min()
df1[['Math','Science','English']].max()
```

Out[20]:

Dept	Math	Science	English
CSE	92.0	91.0	90.0
ECE	74.0	84.0	80.0
ME	69.0	76.0	74.0

Merge

In [21]:
```
attendance = {
    'RollNo': [101, 102, 103, 104, 107],
    'Attendance(%)': [92, 85, 88, 76, 90]
}
```

6. Merge students and attendance on RollNo (inner join).

In [23]:
```
newdf = pd.DataFrame(attendance)
newdf

pd.merge(df,newdf,on='RollNo',how='inner')
```

Out[23]:

	RollNo	Name	Dept	Math	Science	English	Attendance(%)
0	101	Aarav	CSE	88.0	91.0	85.0	92
1	102	Diya	CSE	92.0	NaN	87.0	85
2	103	Ishaan	ECE	NaN	78.0	80.0	88
3	104	Meera	ECE	74.0	84.0	NaN	76

In [24]:
```
sports = {
    'RollNo': [101, 103, 105, 107],
    'Sport': ['Cricket', 'Football', 'Badminton', 'Hockey']
}
```

7. Merge students and sports (outer join) – identify students without sports info.

In [26]:
```
studf = pd.DataFrame(sports)
studf

pd.merge(df,studf,on='RollNo',how='outer')
```

Out[26]:

	RollNo	Name	Dept	Math	Science	English	Sport
0	101	Aarav	CSE	88.0	91.0	85.0	Cricket
1	102	Diya	CSE	92.0	NaN	87.0	NaN
2	103	Ishaan	ECE	NaN	78.0	80.0	Football
3	104	Meera	ECE	74.0	84.0	NaN	NaN
4	105	Kabir	ME	69.0	76.0	74.0	Badminton
5	106	Anaya	CSE	85.0	89.0	90.0	NaN
6	107	NaN	NaN	NaN	NaN	NaN	Hockey

join

8. Convert students and attendance into DataFrames with RollNo as index. Perform a left join on index.

In [30]:
```
pd.merge(df,newdf,on='RollNo',how='left').set_index('RollNo')
```

Out[30]:

RollNo	Name	Dept	Math	Science	English	Attendance(%)
101	Aarav	CSE	88.0	91.0	85.0	92.0
102	Diya	CSE	92.0	NaN	87.0	85.0
103	Ishaan	ECE	NaN	78.0	80.0	88.0
104	Meera	ECE	74.0	84.0	NaN	76.0
105	Kabir	ME	69.0	76.0	74.0	NaN
106	Anaya	CSE	85.0	89.0	90.0	NaN

concat

9. Create a new small DataFrame of newly admitted students:

In [31]:
```
new_students = {
    'RollNo': [109, 110],
    'Name': ['Rohan', 'Sara'],
    'Dept': ['ECE', 'CSE'],
    'Math': [81, 95],
    'Science': [79, 88],
    'English': [83, 91]
}
```

In [32]:
```
newstudf = pd.DataFrame(new_students)
newstudf
```

Out[32]:

	RollNo	Name	Dept	Math	Science	English
0	109	Rohan	ECE	81	79	83
1	110	Sara	CSE	95	88	91

10. Concatenate this DataFrame with the original students.

In [33]:
```
pd.concat([df,newstudf])
```

Out[33]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
1	102	Diya	CSE	92.0	NaN	87.0
2	103	Ishaan	ECE	NaN	78.0	80.0
3	104	Meera	ECE	74.0	84.0	NaN
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0
0	109	Rohan	ECE	81.0	79.0	83.0
1	110	Sara	CSE	95.0	88.0	91.0

11. Concatenate students[['RollNo','Name']] with sports column-wise.

In [ ]:
```
sports = {
    'RollNo': [101, 103, 105, 107],
    'Sport': ['Cricket', 'Football', 'Badminton', 'Hockey']
}
```

In [36]:
```
pd.concat([df[['RollNo', 'Name']], studf[['Sport']]], axis=1)
```

Out[36]:

	RollNo	Name	Sport
0	101	Aarav	Cricket
1	102	Diya	Football
2	103	Ishaan	Badminton
3	104	Meera	Hockey
4	105	Kabir	NaN
5	106	Anaya	NaN

Handle missing value

12. Read one csv file of your choice

Use different techniques to deal with missing values in the file

In [42]:
```
s = pd.read_csv('students.csv')
s
```

Out[42]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
1	102	Diya	CSE	92.0	NaN	87.0
2	103	Ishaan	ECE	NaN	78.0	80.0
3	104	Meera	ECE	74.0	84.0	NaN
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0

In [41]:
```
s.dropna()
```

Out[41]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0

In [44]:
```
s.dropna(axis=1)
```

Out[44]:

	RollNo	Name	Dept
0	101	Aarav	CSE
1	102	Diya	CSE
2	103	Ishaan	ECE
3	104	Meera	ECE
4	105	Kabir	ME
5	106	Anaya	CSE

In [45]:
```
s.dropna(subset=['Math'])
```

Out[45]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
1	102	Diya	CSE	92.0	NaN	87.0
3	104	Meera	ECE	74.0	84.0	NaN
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0

In [46]:
```
s.fillna(0)
```

Out[46]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
1	102	Diya	CSE	92.0	0.0	87.0
2	103	Ishaan	ECE	0.0	78.0	80.0
3	104	Meera	ECE	74.0	84.0	0.0
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0

In [47]:
```
s.fillna(method='ffill')
```

```
C:\Users\Nikhil\AppData\Local\Temp\ipykernel_10532\1029756637.py:1: FutureWarnin
g: DataFrame.fillna with 'method' is deprecated and will raise in a future versio
n. Use obj.ffill() or obj.bfill() instead.
  s.fillna(method='ffill')
```

Out[47]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
1	102	Diya	CSE	92.0	91.0	87.0
2	103	Ishaan	ECE	92.0	78.0	80.0
3	104	Meera	ECE	74.0	84.0	80.0
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0

In [48]:
```
s.fillna(method='bfill')
```

```
C:\Users\Nikhil\AppData\Local\Temp\ipykernel_10532\1641295691.py:1: FutureWarnin
g: DataFrame.fillna with 'method' is deprecated and will raise in a future versio
n. Use obj.ffill() or obj.bfill() instead.
  s.fillna(method='bfill')
```

Out[48]:

	RollNo	Name	Dept	Math	Science	English
0	101	Aarav	CSE	88.0	91.0	85.0
1	102	Diya	CSE	92.0	78.0	87.0
2	103	Ishaan	ECE	74.0	78.0	80.0
3	104	Meera	ECE	74.0	84.0	74.0
4	105	Kabir	ME	69.0	76.0	74.0
5	106	Anaya	CSE	85.0	89.0	90.0

In [51]:
```
s['Science'].fillna(s['Science'].mean())
```

Out[51]:
```
0    91.0
1    83.6
2    78.0
3    84.0
4    76.0
5    89.0
Name: Science, dtype: float64
```

Darshan UNIVERSITY
योग: कर्मसु कौशलम्

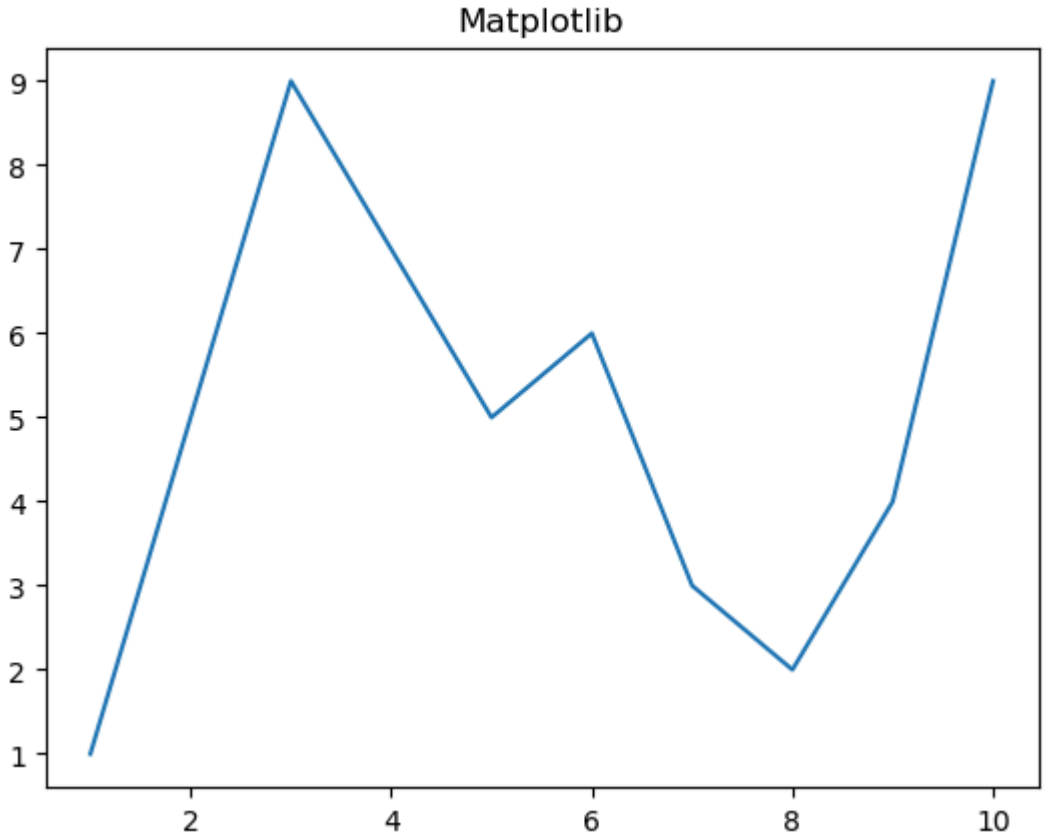
Python for Data Science - 2305CS303

Lab - 12

In [2]:
```
import matplotlib.pyplot as plt
```

In [28]:
```
x = range(1,11)
y = [1,5,9,7,5,6,3,2,4,9]
plt.plot(x,y)
plt.title("Matplotlib")
```

Out[28]: Text(0.5, 1.0, 'Matplotlib')

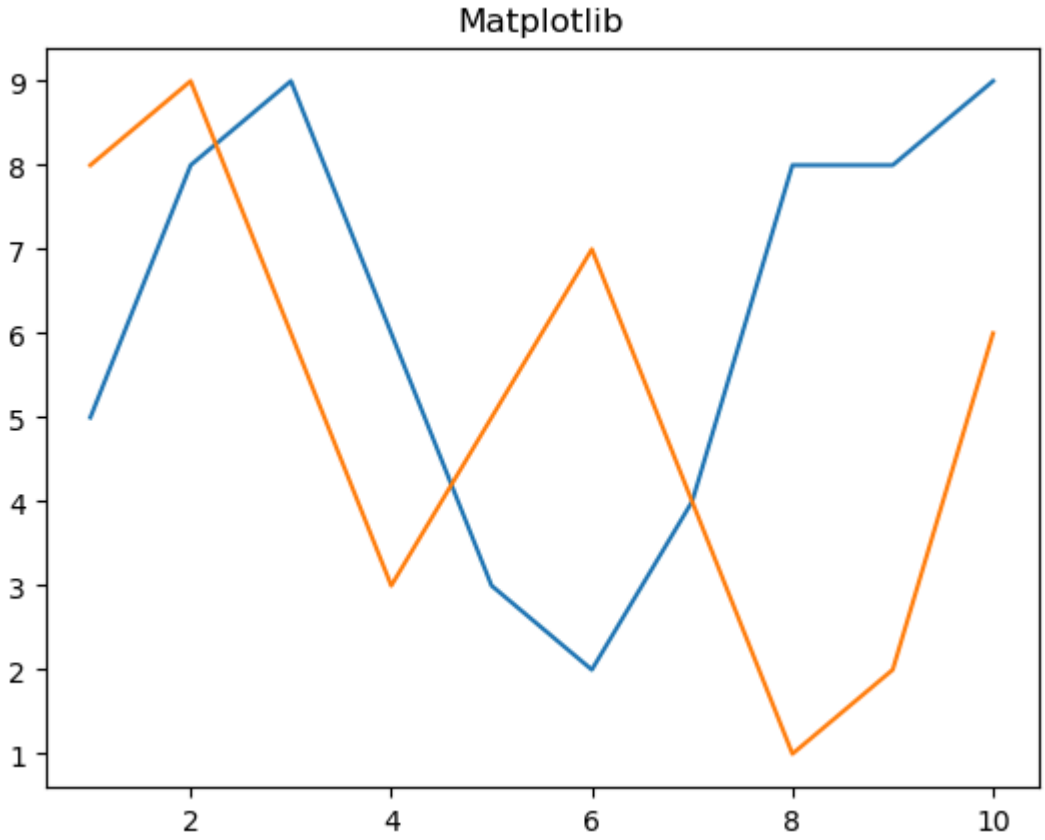


In [29]:
```
x = [1,2,3,4,5,6,7,8,9,10]
cxMarks = [5,8,9,6,3,2,4,8,8,9]
cyMarks = [8,9,6,3,5,7,4,1,2,6]

plt.plot(x,cxMarks)
```

```
plt.plot(x,cyMarks)
plt.title("Matplotlib")
```

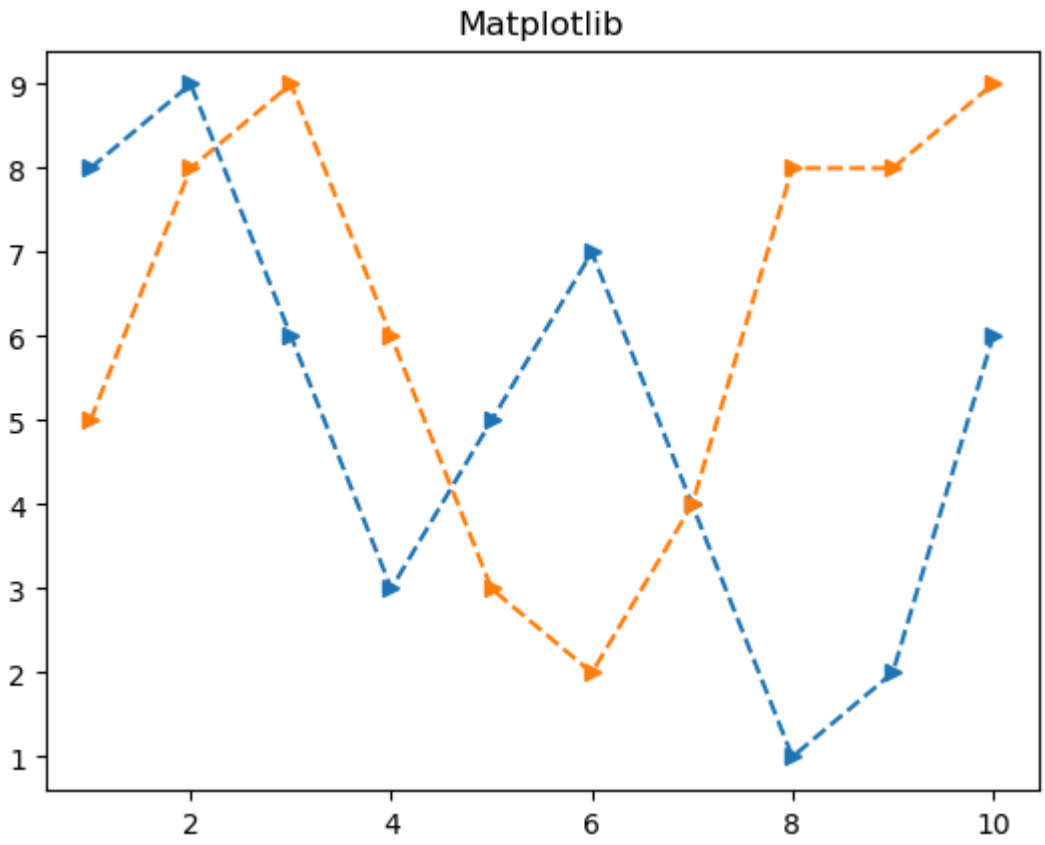
Out[29]: Text(0.5, 1.0, 'Matplotlib')



In [30]:
```
x = range(1,11,1)
cxMarks= [8,9,6,3,5,7,4,1,2,6]
cyMarks= [5,8,9,6,3,2,4,8,8,9]

plt.plot(x,cxMarks,linestyle='dashed',marker='>')
plt.plot(x,cyMarks,linestyle='dashed',marker='>')
plt.title("Matplotlib")
```

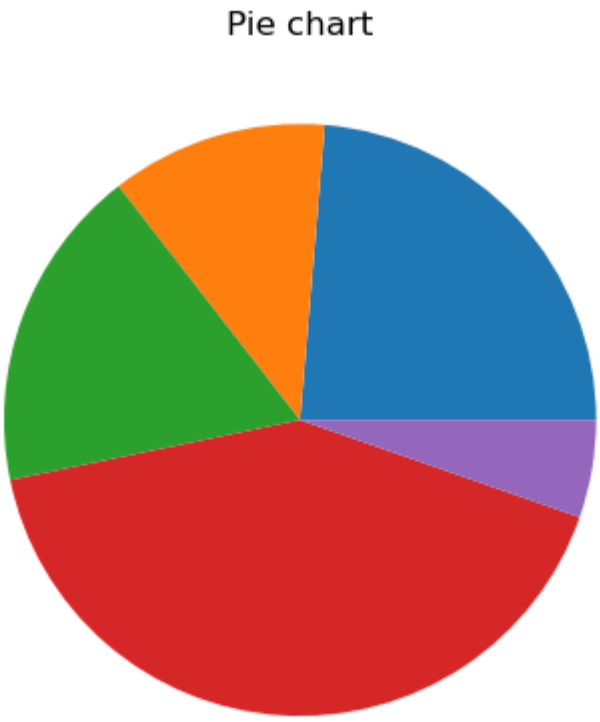
Out[30]: Text(0.5, 1.0, 'Matplotlib')



04) WAP to demonstrate the use of Pie chart.

In [31]:
```
adn = [200,100,150,350,45]
dept = ['MCA','BCA','MBA','BBA','BTECH']
c = ['r','g','b','y','c']
plt.pie(adn)
plt.title("Pie chart")
```

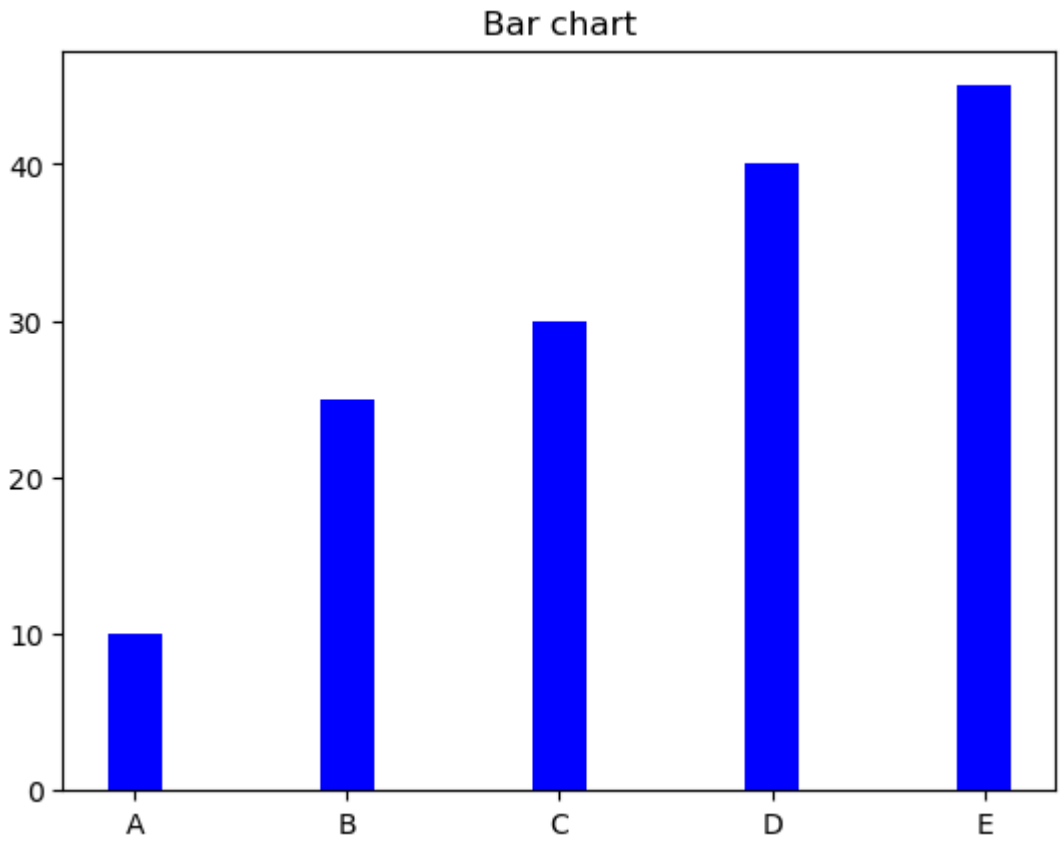
Out[31]:
```
Text(0.5, 1.0, 'Pie chart')
```



05) WAP to demonstrate the use of Bar chart.

In [32]:
```
x = ['A','B','C','D','E']
y = [10,25,30,40,45]
plt.bar(x,y,width=0.25,color='b')
plt.title("Bar chart")
```

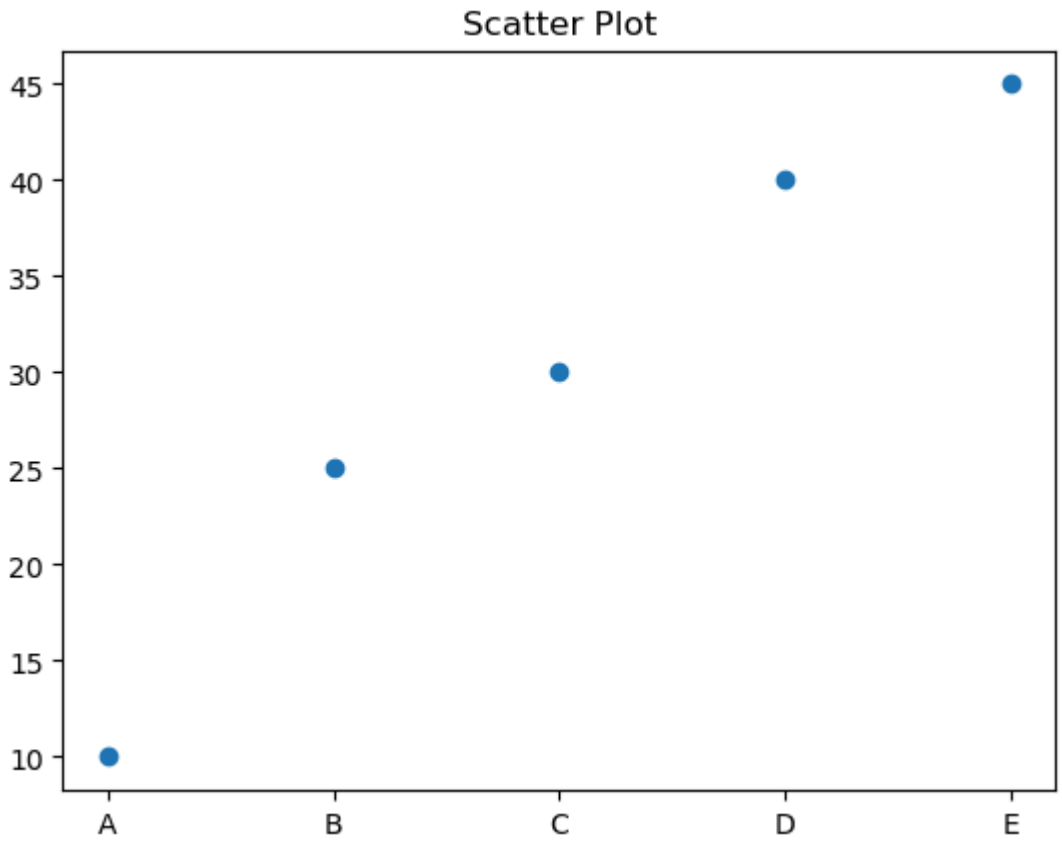
Out[32]:
```
Text(0.5, 1.0, 'Bar chart')
```



06) WAP to demonstrate the use of Scatter Plot.

In [33]:
```
x = ['A','B','C','D','E']
y = [10,25,30,40,45]
plt.scatter(x,y)
plt.title("Scatter Plot")
```

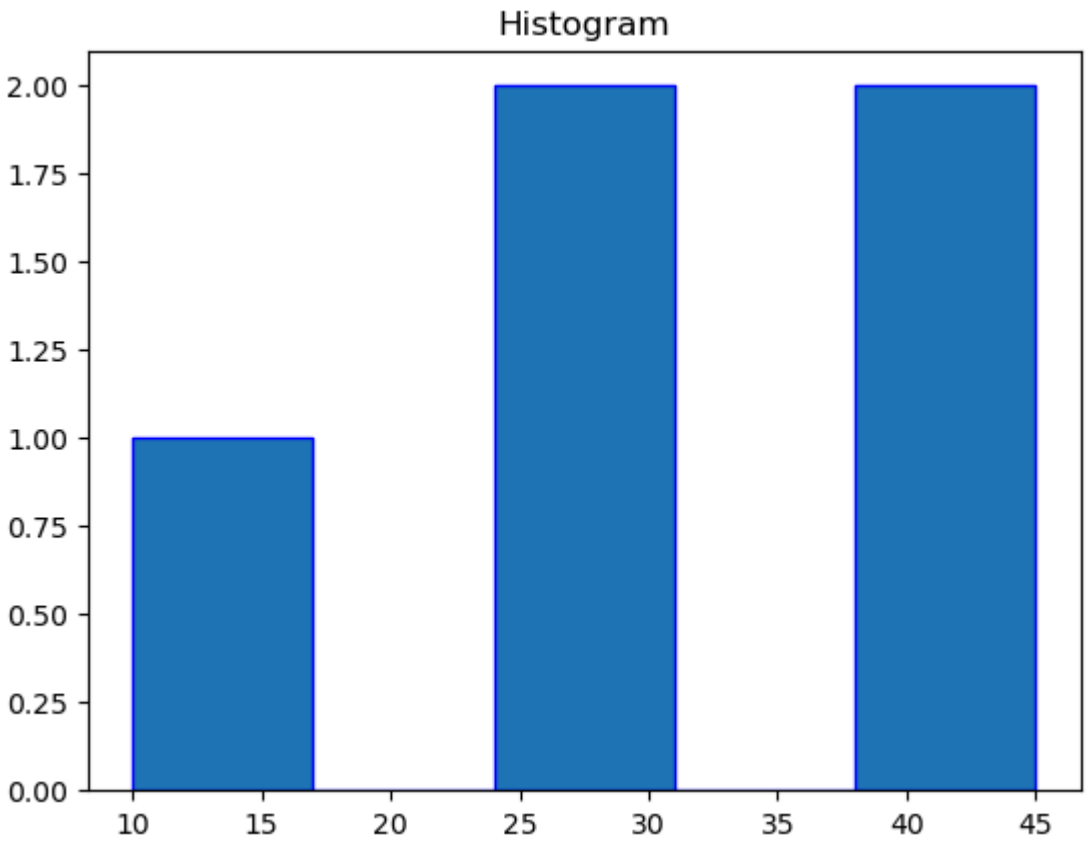
Out[33]:
```
Text(0.5, 1.0, 'Scatter Plot')
```



07) WAP to demonstrate the use of Histogram.

In [34]:
```
x = [10,25,30,40,45]
plt.hist(x,bins=5,edgecolor='b')
plt.title("Histogram")
```

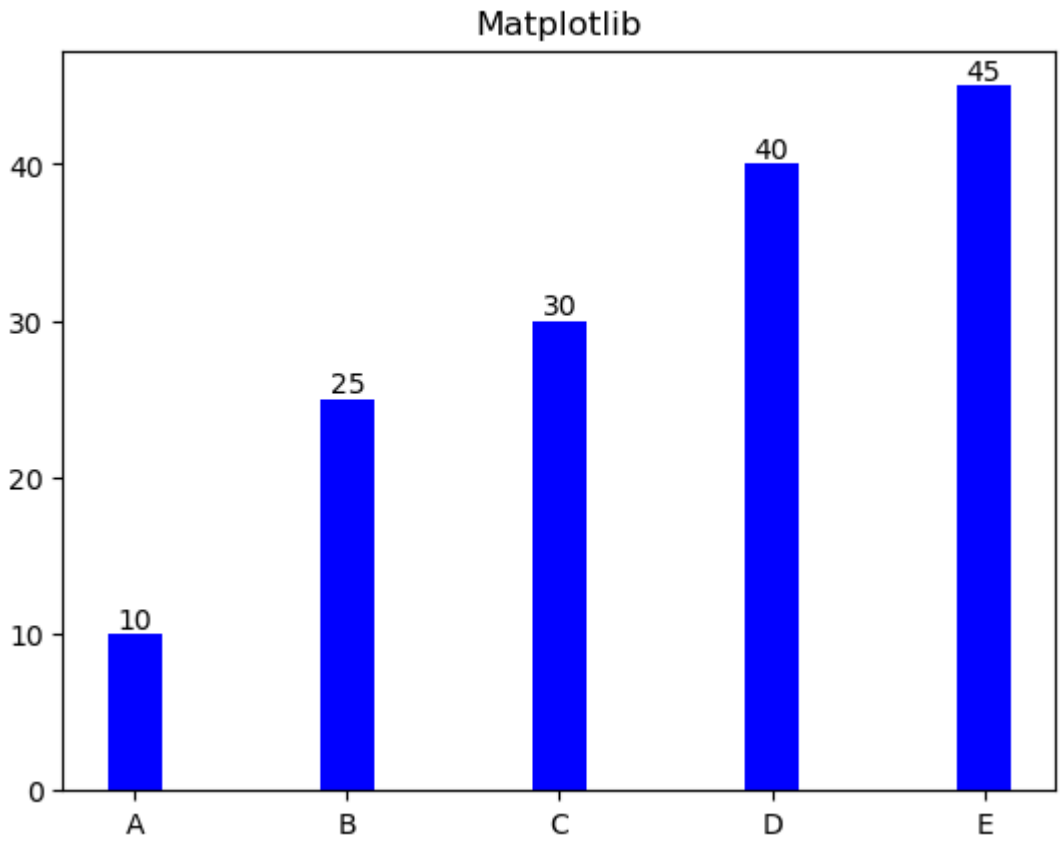
Out[34]:
```
Text(0.5, 1.0, 'Histogram')
```



08) WAP to display the value of each bar in a bar chart using Matplotlib.

In [35]:
```
x = ['A','B','C','D','E']
y = [10,25,30,40,45]
bars = plt.bar(x,y,width=0.25,color='b')
plt.bar_label(bars)
plt.title("Matplotlib")
```

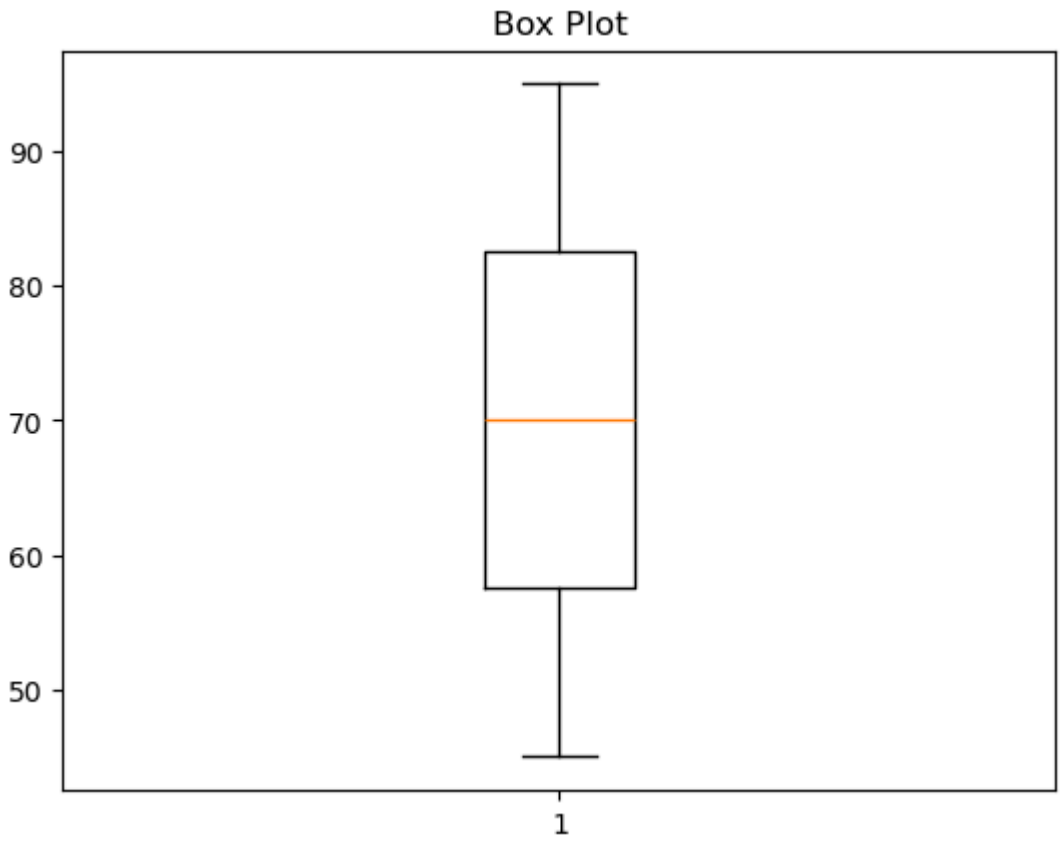
Out[35]:
```
Text(0.5, 1.0, 'Matplotlib')
```



09) WAP to create a Box Plot.

In [36]:
```
data = [45,50,55,60,65,70,75,80,85,90,95]
plt.boxplot(data)
plt.title("Box Plot")
```

Out[36]:
```
Text(0.5, 1.0, 'Box Plot')
```



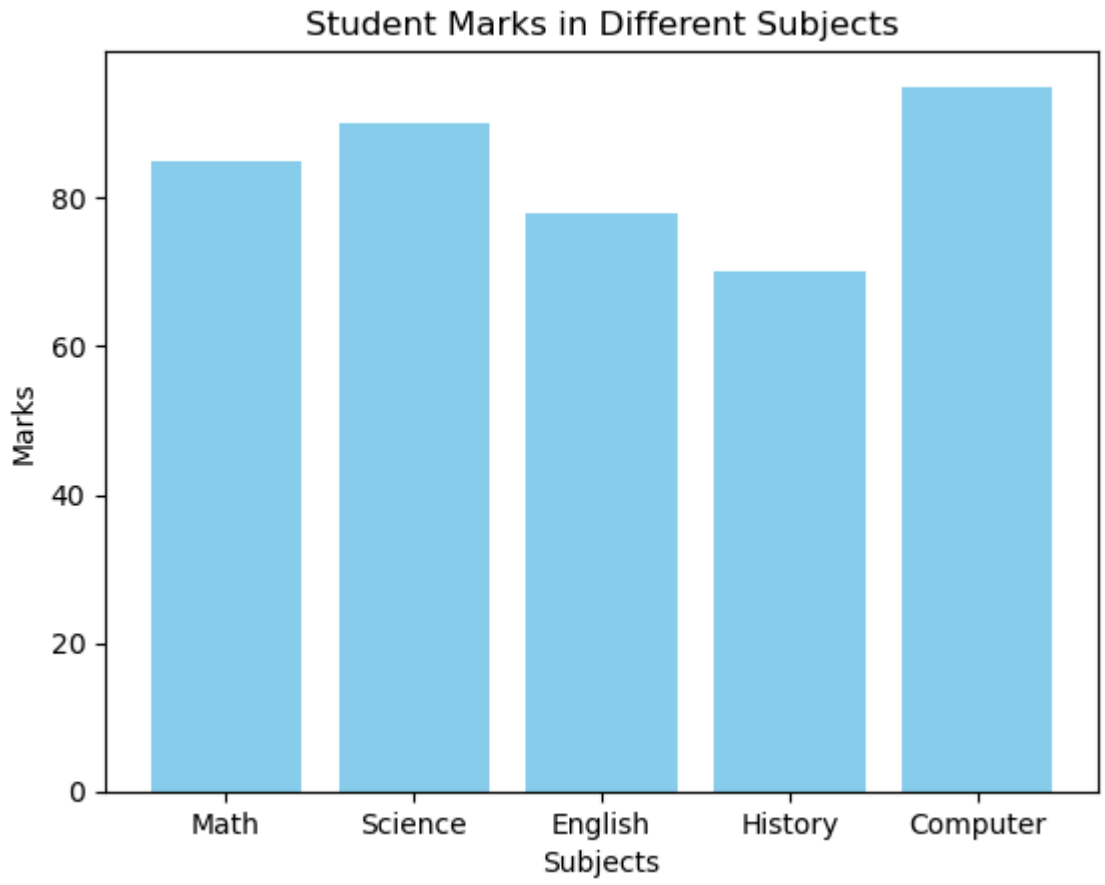
In [27]:
```python
import matplotlib.pyplot as plt

# Sample data
subjects = ["Math", "Science", "English", "History", "Computer"]
marks = [85, 90, 78, 70, 95]

# Create bar chart
plt.bar(subjects, marks, color="skyblue")

# Add labels and title
plt.xlabel("Subjects")
plt.ylabel("Marks")
plt.title("Student Marks in Different Subjects")

# Show chart
plt.show()
```


Student Marks in Different Subjects
Marks
0
20
40
60
80
Math
Science
English
History
Computer
Subjects

In [ ]: