PROGRAM-1

1a) Write a Python program to find the best of two test average marks out of three test marks accepted by the user.

```
   tm1 = int(input("Enter First Test Marks"))
   tm2 = int(input("Enter Second Test Marks"))
   tm3 = int(input("Enter Third Test Marks"))
   if tm1>tm2 and tm1>tm3 :
       if tm2>tm3:
           avg = (tm1+tm2)/2
       else:
           avg = (tm1+tm3)/2
   elif tm2>tm1 and tm2>tm3:
       if tm1>tm3:
           avg = (tm1+tm2)/2
       else:
           avg = (tm2+tm3)/2
   else:
       if tm1>tm2:
           avg = (tm3+tm1)/2
       else:
           avg = (tm3+tm2)/2
print("The average of two best test marks out of three test marks = ",avg)
```

OUTPUT:

```
Enter First Test Marks 85

Enter Second Test Marks 52

Enter Third Test Marks 68

The average of two best test marks out of three test marks = 76.5
```

1b) Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.

```
x = int(input("Enter a number: "))
c0,c1,c2,c3,c4,c5,c6,c7,c8,c9=0,0,0,0,0,0,0,0,0,0
num = x
rev = 0
while x>0:
    r = x%10
    rev = rev*10 + r
    x = x//10
    if r == 0:
        c0+=1
    elif r == 1:
        c1+=1
    elif r == 2:
        c2+=1
    elif r == 3:
        c3+=1
    elif r == 4:
        c4+=1
    elif r == 5:
        c5+=1
    elif r == 6:
        c6+=1
    elif r == 7:
        c7+=1
    elif r == 8:
        c8+=1
    elif r == 9:
        c9+=1
if rev==num:
    print("The Number {0} is palindrome ".format(num))
else:
    print("The Number {0} is not palindrome ".format(num))
print("The occurrence of 0 is {0}, 1 is {1}, 2 is {2}, 3 is {3}, 4 is {4}, 5 is {5}, 6 is {6}, 7 is {7}, 8 is
{8}, 9 is {9}".format(c0,c1,c2,c3,c4,c5,c6,c7,c8,c9))
```

OUTPUT:

1.
```
Enter a number: 141
The Number 141 is palindrome
The occurrence of 0 is 0, 1 is 2, 2 is 0, 3 is 0, 4 is 1, 5 is 0, 6 is 0, 7 is  0, 8 is 0, 9 is 0
```

2.
```
Enter a number: 15698
The Number 15698 is not palindrome
The occurrence of 0 is 0, 1 is 1, 2 is 0, 3 is 0, 4 is 0, 5 is 1, 6 is 1, 7 is 0, 8 is 1, 9 is 1
```

PROGRAM-2

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

```
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:

1.
   ```
   Enter a number : 5
   fn(5) = 3
   ```

2.
   ```
   Enter a number : -4
   Error in input
   ```

2b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

```
def BinToDec(x):
    dec = 0
    i = 0
    while x>0:
        r = x%10
        if r!=0 and r!=1:
            print("Enter a valid Binary number")
            return 0
        else:
            dec = dec + r*2**i
            x = x // 10
            i += 1
    return dec
def OctaToHexa(n):
    num = n
    dec = 0
    base = 1
    temp = num
    while temp:
        r = temp % 10
        temp = temp // 10
        dec += r * base
        base = base * 8
    result = ' '
    while dec != 0:
        temp = 0
        temp = dec % 16
        if temp < 10:
            result = str(temp) + result
        else:
            result = chr(temp + 55) + result
        dec = dec // 16
    return result

x = int(input("Enter a Binary number "))
result = BinToDec(x)
if result:
    print("The Decimal equivalent of {0} is {1}".format(x, result))
y = int(input("Enter a Octal number "))
```

```
result = OctaToHexa(y)
print(result)
if result:
   print("The Hexa Decimal equivalent of {0} is {1}".format(y, result))
```

OUTPUT:

```
Enter a Binary number 1111
The Decimal equivalent of 1111 is 15
Enter a Octal number 146523
CD53
The Hexa Decimal equivalent of 146523 is CD53
```

PROGRAM-3

3a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters, and lowercase letters.

```
x = input("Enter a sentence")
y = x
print("There are",len(x.split())," words in the sentence")
digits,upper,lower=0,0,0
for i in x:
    if i.isdigit():
        digits+=1
    elif i.isupper():
        upper+=1
    elif i.islower():
        lower+=1
print("There are {0} digits, {1} upper case characters and {2} lower case characters in the
sentence".format(digits,upper,lower))
```

OUTPUT:

```
Enter a sentence Rama went to Devaraja market to pick 2 kgs of vegetable
There are 11  words in the sentence
There are 1 digits, 2 upper case characters and 42 lower case characters in the sentence
```

3b) Write a Python program to find the string similarity between two given strings.

```
x = input("Enter first String")
y = input("Enter second String")
x = x.strip()
y = y.strip()
sim=0
if len(x)>len(y):
    xx = x
    yy = y
else:
    xx = y
    yy = x
j=0
for i in yy:
    if i==xx[j]:
        sim+=1
    else:
        pass
    j+=1
similarity = (sim/len(xx))
print("The similarity between the two given strings is", similarity)
```

OUTPUT:

1.
```
Enter first String Python Exercises
Enter second String Python Exercises
The similarity between the two given strings is 1.0
```

2.
```
Enter first String Python Lab
Enter second String Python Laboratory
The similarity between the two given strings is 0.5882352941176471
```

PROGRAM-4

4a) Write a python program to implement insertion sort and merge sort using lists.

```
def InsertionSort(lst):
    i=0
    while i<len(lst):
        small = lst[i]
        for j in range(i+1,len(lst)):
            nxt =lst[j]
            if small>nxt:
                small = lst[j]
        index = lst.index(small)
        if i==index :
            pass
        else:
            lst.remove(small)
            lst.insert(i,small)
        i=i+1
    return lst
def mergeSort(arr):
    if len(arr) > 1:
        mid = len(arr)//2
        sub_array1 = arr[:mid]
        sub_array2 = arr[mid:]
        mergeSort(sub_array1)
        mergeSort(sub_array2)
        i = j = k = 0
        while i < len(sub_array1) and j < len(sub_array2):
            if sub_array1[i] < sub_array2[j]:
                arr[k] = sub_array1[i]
                i += 1
            else:
                arr[k] = sub_array2[j]
                j += 1
            k += 1
        while i < len(sub_array1):
            arr[k] = sub_array1[i]
            i+=1
            k += 1
        while j < len(sub_array2):
            arr[k] = sub_array2[j]
            j += 1
            k += 1
    return arr

lst = []
n = int(input("Enter the size of the list"))
print("Enter ",n," numbers of the list")
for i in range(1,n+1):
    lst.append(int(input()))
print(lst)
print("Enter 1: Insertion Sort, 2: Merge Sort ")
ch = int(input())
if ch==1:
    lst = InsertionSort(lst)
    print("The sorted array is ",lst)
elif ch==2:
    lst = mergeSort(lst)
    print("The sorted array is ",lst)
else:
    print("Invalid Choice")
```

OUTPUT:

1.

```
Enter the size of the list5
Enter 5 numbers of the list
5
4
3
2
1
[5, 4, 3, 2, 1]
Enter 1: Insertion Sort, 2: Merge Sort
1
The sorted array is [1, 2, 3, 4, 5]
```

2.

```
Enter the size of the list5
Enter 5 numbers of the list
10
9
8
7
6
[10, 9, 8, 7, 6]
Enter 1: Insertion Sort, 2: Merge Sort
2
The sorted array is [6, 7, 8, 9, 10]
```

3.

```
Enter the size of the list5
Enter 5 numbers of the list
5
4
3
2
1
[5, 4, 3, 2, 1]
Enter 1: Insertion Sort, 2: Merge Sort
3
Invalid Choice
```

4b) Write a program to convert Roman numbers into integer values using dictionaries

```
def roman_to_int(s):
    rom_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
    int_val = 0
    for i in range(len(s)):
        if i > 0 and rom_val[s[i]] > rom_val[s[i - 1]]:
            int_val += rom_val[s[i]] - 2 * rom_val[s[i - 1]]
        else:
            int_val += rom_val[s[i]]
    return int_val
x = input("Enter the Roman Number")
print(roman_to_int(x.upper()))
```

OUTPUT:

```
Enter the Roman Number XVIII
18
Enter the Roman Number MMMDCCCCLXXXVI
3986
```

PROGRAM-5

5a) Write a function called isphonenumber( ) to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

```
import re
def isphonenumber(x):
    l = len(x)
    if l!=12:
        return 0
    else:
        for i in range(0,l):
            if i==0 or i==1 or i==2 :
                if x[i].isdigit() == False:
                    return 0
            if i==4 or i==5 or i==6 :
                if x[i].isdigit() == False:
                    return 0
            if i==8 or i==9 or i==10 :
                if x[i].isdigit() == False:
                    return 0
            if i==3 or i==7:
                if x[i] != '-':
                    return 0
        return 1
def REisphonenumber(x):
    pno = re.compile(r'\d\d\d-\d\d\d-\d\d\d\d')
    if pno.match(x):
        return 1
    else:
        return 0
phoneNo = input("Enter a phone number of ddd-ddd-dddd format to validate")
ch = int(input("Enter 1.To validate without RE. 2. To validate using RE"))
if ch==1:
    if isphonenumber(phoneNo)==1:
        print("You have entered a valid phone number")
    else:
            print("You have entered an invalid phone number")

elif ch==2:
    if REisphonenumber(phoneNo)==1:
        print("You have entered a valid phone number")
    else:
         print("You have entered an invalid phone number")
```

OUTPUT:

```
1.      Enter a phone number of ddd-ddd-dddd format to validate123-456-7890
        Enter 1.To validate without RE. 2. To validate using RE 1
        You have entered a valid phone number

2.      Enter a phone number of ddd-ddd-dddd format to validate1234567890
        Enter 1.To validate without RE. 2. To validate using RE 1
        You have entered an invalid phone number

3.      Enter a phone number of ddd-ddd-dddd format to validate123-456-7890
        Enter 1.To validate without RE. 2. To validate using RE 2
        You have entered a valid phone number

4.      Enter a phone number of ddd-ddd-dddd format to validate1234567890
        Enter 1.To validate without RE. 2. To validate using RE 2
        You have entered an invalid phone number
```

5b) Develop a Python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com)

```
import re
try:
    file = open("data.txt")
    for line in file:
        line = line.strip()
        match = re.findall(r"(\d{10})", line)
        if(len(match)>0):
            print(match)
        emails = re.findall("[0-9a-zA-z]+@[0-9a-zA-z]+\.[0-9a-zA-z]+", line)
        if(len(emails) > 0):
            print(emails)
except FileNotFoundError as e:
    print(e)
```

Note: Create a text file with a name data.txt in Jupyter Notebook that txt file should contains phone numbers and mail address

OUTPUT:

```
['cmcpc_support@insight.gov']
['1800103421']
['1800103002']
['1800419002']
['8046122000']
['9161464700']
['support765@gmail.com']
```

PROGRAM-6

6a) Write a python program to accept a file name from the user and perform the following operations.

A. Display the first N line of the file

B. Find the frequency of occurrence of the word accepted by the user in the file

```
inputFile=open("data.txt")
N=int(input("enter N Value"))
i=0
while i<=N:
    for line in inputFile:
        print(line)
        i=i+1
count=0
key=input("Enter the word to be searched")
file1=open("data.txt")
for line in file1:
    words=line.split()
    for word in words:
        if word==key:
            count+=1
print("Keyword:",key, "occurs",count, "times")
```

OUTPUT:

```
Enter N value: 16
The following are the first 16 lines of a text file:
Queries related to AIS, TIS, SFT Preliminary response, Response to e-campaigns or e-Verification

09:30 hrs - 18:00 hrs (Monday to Friday)

Email: cmcpc_support@insight.gov.in

Phone2
18001034215

e-Filing of Income Tax Return or Forms and other value added services & Intimation, Rectification
Refund and other Income Tax Processing Related   Queries.

08:00 hrs - 20:00 hrs (Monday to Friday)

09:00 hrs - 18:00 hrs (Saturday's)

Phone2
Enter word to be searched: 18:00
Occurrences of the word:
2
```

6b) Write a Python program to create a ZIP file of a particular folder that contains several filesinside it.

```
from zipfile import ZipFile
import os
from os.path import basename
with ZipFile('F:\SampleDirZipFile.zip', 'w') as zipObj:
    for folderName,subfolders,filenames in os.walk("F:\SampleDir"):
        for filename in filenames:
            print(folderName,subfolders,filename)
            filePath = os.path.join(folderName,filename)
            zipObj.write(filePath, basename(filePath))
print("Created")
```

OUTPUT:

```
F:\SampleDir ['Sample_1', 'Sample_2'] Demo_3.docx
F:\SampleDir\Sample_1\Demo [] Demo_1.docx
F:\SampleDir\Sample_1\Demo [] ~$Demo_1.docx
F:\SampleDir\Sample_2 [] Demo_2.txt
Created
```

PROGRAM-7

7a) By using the concept of inheritance write a python program to find the area of a triangle, circle, and rectangle.

```
class Shape:
    area,radius,length,breadth,a,b,c=0,0,0,0,0,0,0
    def __init__(self, r):
        self.radius = r
    def __init__(self, l,b):
        self.length = l
        self.breadth = b
    def __init__(self, a,b,c):
        self.a = a
        self.b = b
        self.c = c

class Circle(Shape):
    def __init__(self,r):
        super().__init__(r,0,0)

    def area(self,r):
        print("The area of circle is ",3.14*r*r)
class Rectangle(Shape):
    def __init__(self,l,b):
        super().__init__(0,l,b)

    def area(self,l,b):
        print("The area of rectangle is ",l*b)
class Triangle(Shape):
    def __init__(self,a,b,c):
        super().__init__(a,b,c)
    def area(self,a,b,c):
        s = (a+b+c)/2
        import math
        print("The area of triangle is ",math.sqrt(s*(s-a)*(s-b)*(s-c)))

print("Enter your choice to compute area of 1.Circle, 2.Rectangle, 3.Triangle")
ch = int(input())
if ch==1:
    r = int(input("Enter radius of circle"))
```

```
c = Circle(r)
    c.area(r)
elif ch==2:
    l = int(input("Enter length of rectangle"))
    b = int(input("Enter breadth of rectangle"))
    r = Rectangle(l,b)
    r.area(l,b)
elif ch==3:
    a = int(input("Enter length of side a of triangle"))
    b = int(input("Enter length of side b of triangle"))
    c = int(input("Enter length of side c of triangle"))
    t = Triangle(a,b,c)
    t.area(a,b,c)else:
    print("Invalid choice")
```

OUTPUT:

```
1.          Enter your choice to compute area of 1.Circle, 2.Rectangle, 3.Triangle
            1
            Enter radius of circle8
            The area of circle is 200.96


2.          Enter your choice to compute area of 1.Circle, 2.Rectangle, 3.Triangle
            2
            Enter length of rectangle5
            Enter breadth of rectangle7
            The area of rectangle is 35

3.          Enter your choice to compute area of 1.Circle, 2.Rectangle, 3.Triangle
            3
            Enter length of side a of triangle5
            Enter length of side b of triangle6
            Enter length of side c of triangle4
            The area of triangle is 14.696938456699069

4.          Enter your choice to compute area of 1.Circle, 2.Rectangle, 3.Triangle
            4
            Invalid choice
```

7b) Write a Python program by creating a class called Employee to store the details of Name, Employee_ID, Department, and Salary, and implement a method to update the salary of employees belonging to a given department.

```
class Employee:
    def __init__(self,name,eid,dept,sal):
        self.name = name
        self.eid = eid
        self.dept = dept
        self.sal = sal
    def salUpdate(self,eid,dept,updsal):
        self.sal = updsal
if __name__ == "__main__":
    emp=[]
    while True:
        ch = int(input("\n Enter 1.Create Employee\n 2.To display all employees\n 3.To Update an employee
salary\n 4.To exit\n"))
        if ch==1:
            n = input("Employee Name: ")
            i = int(input("Employee ID: "))
            d = input("Employee Department: ")
            s = int(input("Employee Salary: "))
            emp.append(Employee(n,i,d,s))
            print("Employee details created",len(emp))
        elif ch==2:
            for i in emp:
                print("\n Employee Name:{0}\n Employee ID:{1}\n Employee Department:{2}\n Employee
Salary:{3}".format(i.name, i.eid, i.dept, i.sal) )

        elif ch==3:
            upd=0
            print("\n Enter the Department and ID of the employee to update salary")
            empid = int(input("Employee ID: "))
            dept = input("Employee Department: ")
            for i in emp:
                if empid == i.eid and dept == i.dept:
                    upsal = int(input("\n Enter the updated salary"))
                    i.salUpdate(empid,dept,upsal)
                    print("\n Salary updated")

                                                                                                                                    upd=1
```

```
if upd==0:
                 print("\n Employee does not exist")
     elif ch==4:
        break;
     else:
        print("Invalid choice")
```

OUTPUT:

```
Enter 1. Create Employee
          2. To display all employees
          3. To Update an employee salary
          4. To exit
          1
          Employee Name: Anjan
          Employee ID: 001
          Employee Department: CSE
          Employee Salary: 65000
          Employee details created 1

Enter 1. Create Employee
          2. To display all employees
          3. To Update an employee salary
          4. To exit
          1
          Employee Name: Anoop
          Employee ID: 002
          Employee Department: ECE
          Employee Salary: 68000
          Employee details created 2

Enter 1. Create Employee
          2. To display all employees
          3. To Update an employee salary
          4. To exit
          1
          Employee Name: Arun
          Employee ID: 003
          Employee Department: ISE
          Employee Salary: 58000
          Employee details created 3
```

```
 Enter 1. Create Employee
             2. To display all employees
             3. To Update an employee salary
             4. To exit
             2

             Employee Name:Anjan
             Employee ID:1
             Employee Department:CSE
             Employee Salary:65000

 Employee Name:Anoop
             Employee ID:2
             Employee Department:ECE
             Employee Salary:68000

             Employee Name:Arun
             Employee ID:3
             Employee Department:ISE
             Employee Salary:58000

 Enter 1. Create Employee
             2. To display all employees
             3. To Update an employee salary
             4. To exit
             3

             Enter the Department and ID of the employee to update salary
             Employee ID: 003
             Employee Department: ISE

             Enter the updated salary62000

             Salary updated

 Enter 1. Create Employee
             2. To display all employees
             3. To Update an employee salary

4. To exit
                                                                                 2
```

```
        Employee Name:Anjan
        Employee ID:1
        Employee Department:CSE
        Employee Salary:65000

        Employee Name:Anoop
        Employee ID:2
        Employee Department:ECE
        Employee Salary:68000

        Employee Name:Arun
        Employee ID:3
        Employee Department:ISE
        Employee Salary:62000

Enter 1. Create Employee
        2. To display all employees
        3. To Update an employee salary
        4. To exit
        4
```

PROGRAM-8

8. Write a Python program to find whether the given input is palindrome or not (for both stringand integer) using the concept of polymorphism and inheritance.

```
class strPalindrome:
    def __init__(self):
        self.word=""
        self.ll=""
    def check(self,s):
        self.word = list(s)
        ll=self.word.copy()
        self.word.reverse()
        if (ll==self.word):
            print("\nIt is Palindrome")
        else:
            print("\nIt is Not Palindrome")
class noPalindrome(strPalindrome):
    def __init__(self):
        super().__init__
    def check(self,no):
        super().check(str(no))
if __name__=='__main__':
    while True:
        ch = int(input("Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit : "))
        if ch==1:
            text = input("Enter a string to check : ")
            s = strPalindrome()
            s.check(text)
        elif ch==2:
            text = int(input("Enter a number to check : "))
            s = noPalindrome()
            s.check(text)
        elif ch==3:
            break
        else:
            print("Invalid choice")
```

OUTPUT:

```
Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit: 1
Enter a string to check: MADAM

It is Palindrome
Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit: 1
Enter a string to check: HELLO

It is Not Palindrome
Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit: 2
Enter a number to check: 121

It is Palindrome
Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit: 2
Enter a number to check: 123

It is Not Palindrome
Enter 1.For String palindrome 2.For Integer Palindrome 3.To Exit: 3
```

PROGRAM-9

9a) Write a Python program to download all XKCD comics.

```
import requests, os, bs4
url = 'http://xkcd.com'
os.makedirs('xkcd',exist_ok = True)
while not url.endswith("#"):
    print("Downloading the page ... ")
    res = requests.get(url)
    res.raise_for_status()
    try:
        soup = bs4.BeautifulSoup(res.text,'lxml')
    except bs4.FeatureNotFound:
        soup = bs4.BeautifulSoup(res.text,'html.parser')

    comic_element = soup.select('#comic img')
    if comic_element == []:
        print("No comic image found!!..")
    else:
        comic_image_url = comic_element[0].get('src')
        print("Downloading the image %s .. " %(comic_image_url))
        res = requests.get('http:' + comic_image_url)
        res.raise_for_status()
        file = open( os.path.join('xkcd',os.path.basename(comic_image_url)) , 'wb')
        for chunk in res.iter_content(10000):
            file.write(chunk)
        file.close()

    prev_link = soup.select('a[rel="prev"]')[0]
    url = 'http://xkcd.com' + prev_link.get('href')
print("Done")
```

OUTPUT:

```
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/alphabet_notes.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/garden_path_sentence.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/summer_solstice.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/bookshelf_sorting.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/heat_pump.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/making_plans.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/musical_scales.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/iceberg.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/ufo_evidence.png ..
Downloading the page ...
Downloading the image //imgs.xkcd.com/comics/marble_run.png ..
```

9b) Demonstrate a python program to read the data from the spreadsheet and write the data intothe spreadsheet.

```
from openpyxl import Workbook
from openpyxl.styles import Font

wb = Workbook()
sheet = wb.active
sheet.title = "Language"
wb.create_sheet(title = "Capital")

lang = ["Kannada", "Telugu", "Tamil"]
state = ["Karnataka", "Telangana", "Tamil Nadu"]
capital = ["Bengaluru", "Hyderabad", "Chennai"]
code =['KA', 'TS', 'TN']

sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Language"
sheet.cell(row = 1, column = 3).value = "Code"

ft = Font(bold=True)
for row in sheet["A1:C1"]:
    for cell in row:
        cell.font = ft

for i in range(2,5):
    sheet.cell(row = i, column = 1).value = state[i-2]
    sheet.cell(row = i, column = 2).value = lang[i-2]
    sheet.cell(row = i, column = 3).value = code[i-2]

wb.save("F:\demo.xlsx")
```

```
sheet = wb["Capital"]

sheet.cell(row = 1, column = 1).value = "State"
sheet.cell(row = 1, column = 2).value = "Capital"
sheet.cell(row = 1, column = 3).value = "Code"

ft = Font(bold=True)
for row in sheet["A1:C1"]:
    for cell in row:
        cell.font = ft

for i in range(2,5):
    sheet.cell(row = i, column = 1).value = state[i-2]
    sheet.cell(row = i, column = 2).value = capital[i-2]
    sheet.cell(row = i, column = 3).value = code[i-2]

wb.save("demo.xlsx")
srchCode = input("Enter state code for finding capital ")
for i in range(2,5):
    data = sheet.cell(row = i, column = 3).value
    if data == srchCode:
        print("Corresponding capital for code", srchCode, "is", sheet.cell(row = i, column = 2).value)

sheet = wb["Language"]
srchCode = input("Enter state code for finding language ")
for i in range(2,5):
    data = sheet.cell(row = i, column = 3).value
    if data == srchCode:
        print("Corresponding language for code", srchCode, "is", sheet.cell(row = i, column = 2).value)
wb.close()
```

OUTPUT:

```
Enter state code for finding capital TN
Corresponding capital for code TN is Chennai
Enter state code for finding language KA
Corresponding language for code KA is Kannada
```

PROGRAM-10

10a) Write a Python program to combine select pages from many PDFs.

```
import PyPDF2
def merge_pdf_files(file1, file2, output_file,file1_pages,file2_pages):
    with open(file1, 'rb') as file1_obj, open(file2, 'rb') as file2_obj:
        pdf1 = PyPDF2.PdfReader(file1_obj)
        pdf2 = PyPDF2.PdfReader(file2_obj)
        pdf_writer = PyPDF2.PdfWriter()
        for page_num in range(file1_pages):
            page = pdf1.pages[page_num]
            pdf_writer.add_page(page)
        for page_num in range(file2_pages):
            page = pdf2.pages[page_num]
            pdf_writer.add_page(page)
        with open(output_file, 'wb') as output:
            pdf_writer.write(output)
        print(f"The PDF files '{file1}' and '{file2}' have been merged into '{output_file}'.")

pdf_file1 = 'F:\Python\sample3.pdf'
pdf_file2 = 'F:\Python\sample4.pdf'
output_pdf = 'F:\Python\merged.pdf'
file1_pages=int(input("Enter the number of file 1 pages"))
file2_pages=int(input("Enter the number of file 2 pages"))

merge_pdf_files(pdf_file1, pdf_file2, output_pdf,file1_pages,file2_pages)
```

OUTPUT:

```
 Enter the number of file 1 pages2
 Enter the number of file 2 pages4
The PDF files 'F:\Python\sample3.pdf' and 'F:\Python\sample4.pdf' have been merged into
'F:\Python\merged.pdf'.
```

10b) Write a Python program to fetch current weather data from the JSON file.

```
import requests
res = requests.get('https://ipinfo.io/')
data = res.json()
citydata = data['city']
print(citydata)
url = 'https://wttr.in/{}'.format(citydata)
res = requests.get(url)
print(res.text)
```

OUTPUT:

```
Bengaluru
Weather report: Bengaluru

     \  /       Partly cloudy
   _ /"".-.     +25(26) °C
     \_(   ).   → 19 km/h
     /(___(__)  6 km
                0.0 mm
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Wed 12 Jul
├───────────────────────┬──────────────────────────────┤
│            Morning           │             Noon      └──────┬──────┘     Evening
│             Night            │                              │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────
──────────┼──────────────────────────────┤
│    \  /       Partly cloudy  │  _`/"".-.     Patchy rain po…│      .-.
Patchy light r…│  _`/"".-.     Moderate or he…│
│  _ /"".-.     +24(26) °C     │   ,\_(   ).   +28(29) °C     │     (   ).
+24(25) °C     │   ,\_(   ).   +22(25) °C     │
│    \_(   ).   → 20-23 km/h   │    /(___(__)  → 21-26 km/h   │    (___(__)   → 22-
33 km/h   │    /(___(__)  → 13-19 km/h   │
│    /(___(__)  10 km          │      ʻ ʻ ʻ ʻ  9 km           │     ʻ ʻ ʻ ʻ   9 km
│    ‚ʻ‚ʻ‚ʻ‚ʻ   7 km           │
│               0.0 mm | 0%    │     ʻ ʻ ʻ ʻ   0.4 mm | 85%   │    ʻ ʻ ʻ ʻ    0.9
mm | 87%   │    ‚ʻ‚ʻ‚ʻ‚ʻ   2.6 mm | 74%   │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────
──────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Thu 13 Jul
├───────────────────────┬──────────────────────────────┤
│            Morning           │             Noon      └──────┬──────┘     Evening
│             Night            │                              │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────
──────────┼──────────────────────────────┤
```

```
│     \   /          Partly cloudy    │    _`/"".-.      Patchy rain po…│    _`/"".-.      Light
rain sho…│    _`/"".-.       Light rain sho…│
│    _ /"".-.        +24(25) °C        │     ,\_(    ).    +27(29) °C      │     ,\_(    ).
+25(26) °C       │     ,\_(    ).   +22(25) °C      │
│     \_(    ).      ↗ 19-23 km/h      │     /(___(__)    ↗ 19-24 km/h     │     /(___(__)    → 19-
27 km/h     │     /(___(__)    → 20-35 km/h    │
│     /(___(__)      10 km             │       ‘ ‘ ‘ ‘    9 km             │       ‘ ‘ ‘ ‘    10 km
│       ‘ ‘ ‘ ‘      10 km             │
│                    0.0 mm | 0%       │      ‘ ‘ ‘ ‘     0.5 mm | 70%     │      ‘ ‘ ‘ ‘     1.7
mm | 60%     │     ‘ ‘ ‘ ‘       1.0 mm | 61%     │
└──────────────────────────────────────┴──────────────────────────────────┴──────────────────────
──────────┴──────────────────────────────────────┘
                                                                 ┌─────────────┐
┌──────────────────────────────────────┬──────────────────────────┤  Fri 14 Jul  │
│──────────────────────────────┬───────│──────────────────────────┤             │
│            Morning                   │          Noon            └──────┬──────┘      Evening
│             Night                    │                                 │
├──────────────────────────────────────┼─────────────────────────────────┼──────────────────────
──────────┼──────────────────────────────────────┤
│                    Overcast          │                  Cloudy         │    _`/"".-.      Light
rain sho…│    _`/"".-.       Light rain sho…│
│      .--.          +24(25) °C        │      .--.        +27(28) °C      │     ,\_(    ).
+25(27) °C       │     ,\_(    ).   +23(25) °C      │
│   .-(    ).        → 24-28 km/h      │   .-(    ).      → 23-27 km/h     │     /(___(__)    → 17-
24 km/h     │     /(___(__)    → 21-34 km/h    │
│  (___.__)__)       10 km             │  (___.__)__)     10 km            │       ‘ ‘ ‘ ‘    10 km
│       ‘ ‘ ‘ ‘      10 km             │
│                    0.0 mm | 0%       │                  0.0 mm | 0%      │      ‘ ‘ ‘ ‘     1.5
mm | 62%     │     ‘ ‘ ‘ ‘       0.8 mm | 60%     │
└──────────────────────────────────────┴─────────────────────────────────┴──────────────────────
──────────┴──────────────────────────────────────┘
Location: Bengaluru, Bangalore Urban, Karnataka, 560001, India
[12.9791198,77.5912997]

Follow @igor_chubin for wttr.in updates
```

VIVA QUESTIONS

What is Python?

1. Python is one of the most widely-used and popular programming languages, was developed by Guido van Rossum and released first on February 20, 1991.
2. Python is a free and open-source language with a very simple and clean syntax which makes it easy for developers to learn Python.
3. It supports object-oriented programming and is most commonly used to perform general-purpose programming.
4. Python is used in several domains like Data Science, Machine Learning, Deep Learning, Artificial Intelligence, Scientific Computing Scripting, Networking, Game Development Web Development, Web Scraping, and various other domains, System Scripting, Software Development, and Complex Mathematics.

What are the benefits of using Python language as a tool in the present scenario?

The following are the benefits of using Python language:

Object-Oriented Language, High-Level Language, Dynamically Typed language, Extensive support Libraries, Presence of third-party modules, Open source and community development, Portable and Interactive, Portable across Operating systems.

Is Python a compiled language or an interpreted language?

Python is a partially compiled language and partially interpreted language. '#' is used to comment on everything that comes after on the line.

Difference between a Mutable datatype and an Immutable data type?

Mutable data types can be edited i.e., they can change at runtime. Eg – List, Dictionary, etc. Immutable data types can not be edited i.e., they can not change at runtime. Eg – String, Tuple, etc.

What is a lambda function?

A lambda function is an anonymous function. This function can have any number of parameters but,can have just one statement.

Pass means performing no operation or in other words, it is a placeholder in the compound statement,wher there should be a blank left and nothing has to be written there.

Python provides various web frameworks to develop web applications.

The popular python web frameworks are Django, Pyramid, Flask.

- Python's standard library supports for E-mail processing, FTP, IMAP, and other Internet protocols.
- Python's SciPy and NumPyhelp in scientific and computational application development.
- Python's Tkinter library supports to create desktop-based GUI applications.
-

What is the difference between / and // in Python?

// represents floor division whereas / represents precise division. 5//2 = 2 5/2 = 2.5

Yes, **indentation is required in Python**. A Python interpreter can be informed that a group of statements belongs to a specific block of code by using Python indentation. Indentations make the code easy to read for developers in all programming languages but in Python, it is very important to indent the code in a specific order.

What is Scope in Python?

The location where we can find a variable and also access it if required is called the scope of a variable.

Python Local variable: Local variables are those that are initialized within a function and are unique to that function. It cannot be accessed outside of the function.

Python Global variables: Global variables are the ones that are defined and declared outside any function and are not specified to any function.

Module-level scope: It refers to the global objects of the current module accessible in the program.

Outermost scope: It refers to any built-in names that the program can call. The name referenced is located last among the objects in this scope.

Python documentation strings(or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods.

Declaring Docstrings: The docstrings are declared using ”’triple single quotes”’ or “””triple double quotes””” just below the class, method, or function declaration. All functions should have a docstring.

Accessing Docstrings: The docstrings can be accessed using the __doc__ method of the object or

using the help function.

What is slicing in Python?

Python Slicing is a string operation for extracting a part of the string, or some part of a list. With this operator, one can specify where to start the slicing, where to end, and specify the step. List slicing returns a new list from the existing list.

PIP is an acronym for Python Installer Package which provides a seamless interface to install various Python modules. It is a command-line tool that can search for packages over the internet and install them without any user interaction.

Sample Programs

1. Python program to print "Hello Python"

```
print ('Hello Python')
```

2. Python program to do arithmetical operations

```
# Store input numbers:
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')
# Add two numbers
sum = float(num1) + float(num2)
# Subtract two numbers
min = float(num1) - float(num2)
# Multiply two numbers
mul = float(num1) * float(num2)
#Divide two numbers
div = float(num1) / float(num2)
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
# Display the subtraction
print('The subtraction of {0} and {1} is {2}'.format(num1, num2, min))
# Display the multiplication
print('The multiplication of {0} and {1} is {2}'.format(num1, num2, mul))
# Display the division
print('The division of {0} and {1} is {2}'.format(num1, num2, div))
```

3. Python program to find the area of a triangle

```
a = 5
b = 6
c = 7
# Uncomment below to take inputs from the user
# a = float(input('Enter first side: '))
# b = float(input('Enter second side: '))
# c = float(input('Enter third side: '))
# calculate the semi-perimeter
s = (a + b + c) / 2
# calculate the area
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
print('The area of the triangle is %0.2f' %area)
```

4. Python program to swap two variables

```
#swapping of 2 numbers
p=int(input("enter the value of a"))
q=int(input("enter the value of b"))
print('Before Swapping the value of p=',p,'and q=',q)
temp=p
p=q
q=temp
print('After swapping the value of p=',p,'and q=',q)
```

5. Python program to find the sum and average of natural numbers up to n where n is provided by user.

```
n=int(input("Enter upto which number you want sum and average"))
sum=0
for i in range(0,n+1):
sum=sum+i
avg=sum/n
print("Result of sum is",sum)
print("Result of Average",avg)
```

6. WAP to find Factorial of a number using for loop

```
fact=1
n=int(input("enter the value of n to find factorial of a given number"))
                                                                        for i in range(1,n+1) :
```

```
fact=fact*i
print(fact)
```

7. WAP to find Factorial of a number using while loop

```
fact=1
i=1
n=int(input("enter the value of n to find factorial of a given number"))
while i<=n :
fact=fact*i
i=i+1
print(fact)
```

8. WAP to find fibonacci series using Iterative:

```
n = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0
if n <= 0:
print("Please enter a positive integer")
elif n == 1:
print("Fibonacci sequence upto",n,":")
print(n1)
else:
print("Fibonacci sequence:")
while count < n:
print(n1)
next = n1 + n2
n1 = n2
n2 = next
count += 1
```

9.WAP to find fibonacci series using recursion:

```
def fib(n):
if n <= 1:
return n
else:
return(fib(n-1) + fib(n-2))
nterms = int(input("How many terms? "))
if nterms<= 0:
print("Plese enter a positive integer")
else:
print("Fibonacci sequence:")
for i in range(nterms):
print(fib(i))
```

10 . WAP to find largest among three numbers, input by user

```
n1=int(input("enter first number"))
n2=int(input("enter sec number"))
n3=int(input("enter third number"))
if n1>n2 and n1>n3 :
print("n1 is larger")
elif n2>n3 and n2>n1 :
print("n2 is larger")
else :
print("n3 is larger")
```

11. WAP to print first ten programs using for loop .

```
for i in range(1,11) :
print(i)
```

12. WAP to print first ten programs using while loop.

```
I=1
                                                                                while i<=10:
```

```
print(i)
i=i+1
```

13. WAP to check whether a person is eligible for voting.

```
age=input()
type(age)
x=int(age)
if x>18 :
print("eligible")
else :
print("not ")
```

14. **WAP to print grades obtained by the students and print the appropriate message.**

```
marks=input()
type(marks)
x=int(marks)
if x>=90 and x<100:
print('distinction')
elif x>=80 and x<=90:
print("first")
else :
print("fail")
```

15.WAP to find factorial of given number using for loop.

```
fact=1
for i in range(1,6) :
fact*=i
print(fact)
```

16.WAP to find factorial of given number using while loop.

```
fact=1
                                                  i=1
```

```
n=int(input())
while i<=n:
fact=fact*i
i=i+1
print(fact)
```

17. WAP to find factorial of given number using functions.

```
def fact(n) :
fact=1
i=1
while i<=n :
fact*=i
i=i+1
return fact
#print(fact)
n=int(input("enter the number to find factoral of a given number"))
print(fact(n))
```

18.WAP to find gcd of 2 numbers.

```
def gcd(a,b):
if b==0:
return a
else :
return gcd(b,a%b)
n1=int(input("enter the first number"))
n2=int(input("enter the second number"))
print(gcd(n1,n2))
```

19. Python program to generate a random number

20. Python program to convert kilometers to miles.

21. Python program to convert Celsius to Fahrenheit

22. Python program to display calendar
23. Python Program to Check if a Number is Positive, Negative or Zero
24. Python Program to Check if a Number is Odd or Even
25. Python Program to Check Leap Year
26. Python Program to Check Prime Number
27. Python Program to Print all Prime Numbers in an Interval
28. Python Program to Find the Factorial of a Number
29. Python Program to Display the multiplication Table
30. Python Program to Print the Fibonacci sequence
31.Python Program to Check Armstrong Number
32.Python Program to Find Armstrong Number in an Interval

Python Additional Programs

1. Python program to print "Hello Python"
2. Python program to do arithmetical operations
3. Python program to find the area of a triangle
4. Python program to solve quadratic equation
5. Python program to swap two variables
6. Python program to generate a random number
7. Python program to convert kilometers to miles
8. Python program to convert Celsius to Fahrenheit
9. Python program to display calendar
10. Python Program to Check if a Number is Positive, Negative or Zero
11. Python Program to Check if a Number is Odd or Even
12. Python Program to Check Leap Year
13. Python Program to Check Prime Number
14. Python Program to Print all Prime Numbers in an Interval
15. Python Program to Find the Factorial of a Number
16. Python Program to Display the multiplication Table
17. Python Program to Print the Fibonacci sequence
18. Python Program to Check Armstrong Number
19. Python Program to Find Armstrong Number in an Interval

20. Python Program to Find the Sum of Natural Numbers
21. Python Function Programs
22. Python Program to Find LCM
23. Python Program to Find HCF
24. Python Program to Convert Decimal to Binary, Octal and Hexadecimal
25. Python Program To Find ASCII value of a character
26. Python Program to Make a Simple Calculator
27. Python Program to Display Calendar
28. Python Program to Display Fibonacci Sequence Using Recursion
29. Python Program to Find Factorial of Number Using Recursion
30. Python Number Programs
31. Python program to check if the given number is a Disarium Number
32. Python program to print all disarium numbers between 1 to 100
33. Python program to check if the given number is Happy Number
34. Python program to print all happy numbers between 1 and 100
35. Python program to determine whether the given number is a Harshad Number
36. Python program to print all pronic numbers between 1 and 100
37. Python Array Programs
38. Python program to copy all elements of one array into another array
39. Python program to find the frequency of each element in the array
40. Python program to left rotate the elements of an array
41. Python program to print the duplicate elements of an array
42. Python program to print the elements of an array
43. Python program to print the elements of an array in reverse order
44. Python program to print the elements of an array present on even position
45. Python program to print the elements of an array present on odd position
46. Python program to print the largest element in an array
47. Python program to print the smallest element in an array
48. Python program to print the number of elements present in an array
49. Python program to print the sum of all elements in an array
50. Python program to right rotate the elements of an array

51. Python program to sort the elements of an array in ascending order
52. Python program to sort the elements of an array in descending order
53. Python Matrix Programs
54. Python Program to Add Two Matrices
55. Python Program to Multiply Two Matrices
56. Python Program to Transpose a Matrix
57. Python String Programs
58. Python Program to Sort Words in Alphabetic Order
59. Python Program to Remove Punctuation From a String
60. Python Program to reverse a string
61. Python Program to convert list to string
62. Python Program to convert int to string
63. Python Program to concatenate two strings
64. Python Program to generate a Random String
65. Python Program to convert Bytes to string
66. Python List Programs
67. Python Program to append element in the list
68. Python Program to compare two lists
69. Python Program to convert list to dictionary
70. Python Program to remove an element from a list
71. Python Program to add two lists
72. Python Program to convert List to Set
73. Python Program to convert list to string
74. Python Dictionary Programs
75. Python Program to create a dictionary
76. Python Program to convert list to dictionary
77. Python Program to sort a dictionary
78. Python Program to Merge two Dictionaries
79. Python Searching and Sorting Programs
80. Binary Search in Python
81. Linear Search in Python
82. Bubble Sort in Python

83. Insertion Sort in Python
84. Heap Sort in Python
85. Merge Sort in Python
86. Python Circular Linked List Programs
87. Python program to create a Circular Linked List of N nodes and count the number of nodes
88. Python program to create a Circular Linked List of n nodes and display it in reverse order
89. Python program to create and display a Circular Linked List
90. Python program to delete a node from the beginning of the Circular Linked List
91. Python program to delete a node from the end of the Circular Linked List
92. Python program to delete a node from the middle of the Circular Linked List
93. Python program to find the maximum and minimum value node from a circular linked list
94. Python program to insert a new node at the beginning of the Circular Linked List
95. Python program to insert a new node at the end of the Circular Linked List
96. Python program to insert a new node at the middle of the Circular Linked List
97. Python program to remove duplicate elements from a Circular Linked List
98. Python program to search an element in a Circular Linked List
99. Python program to sort the elements of the Circular Linked List
100. Python Doubly Linked List Programs
101. Python program to convert a given binary tree to doubly linked list
102. Python program to create a doubly linked list from a ternary tree
103. Python program to create a doubly linked list of n nodes and count the number of nodes
104. Python program to create a doubly linked list of n nodes and display it in reverse order
105. Python program to create and display a doubly linked list
106. Python program to delete a new node from the beginning of the doubly linked list
107. Python program to delete a new node from the end of the doubly linked list
108. Python program to delete a new node from the middle of the doubly linked list

109. Python program to find the maximum and minimum value node from a doubly linked list

110. Python program to insert a new node at the beginning of the Doubly Linked list

111. Python program to insert a new node at the end of the Doubly Linked List

112. Python program to insert a new node at the middle of the Doubly Linked List

113. Python program to remove duplicate elements from a Doubly Linked List

114. Python program to rotate doubly linked list by N nodes

115. Python program to search an element in a doubly linked list