PART A, PGM 1 – Check if a number belongs to the Fibonacci Sequence.

```
import math

def is_perfect_square(x):
    s = int(math.sqrt(x))
    return s * s == x

def is_fibonacci(n):
    return is_perfect_square(5 * n * n + 4) or is_perfect_square(5 * n * n - 4)

# Example usage:
number = int(input("enter a number"))
if is_fibonacci(number):
    print(number,"is a Fibonacci number.")
else:
    print(f"{number} is not a Fibonacci number.")
```

PART A, PGM 2 – Solve Quadratic Equations

```
import math
a=int(input("Enter the value of a:"))
b=int(input("Enter the value of b:"))
c=int(input("Enter the value of c:"))
if a==0:
    print("Enter correct quadratic value")
else:
    disc=b*b-4*a*c
    sqrt_val=math.sqrt(abs(disc))

    if disc>0:
        print("Real and distinct Roots.")
        print((-b+sqrt_val)/(2*a))
        print((-b-sqrt_val)/(2*a))
    elif disc==0:
        print("Real and Equal Roots.")
        print(-b/(2*a))
    else:            #disc<0
        print("Not Real and Imaginary Roots.")
        print(-b/(2*a),"+j",sqrt_val)
        print(-b/(2*a),"-j",sqrt_val)
```

PART A, PGM 3 – Find the sum of n natural numbers.

```
def sum_of_natural_numbers(N):
        total = 0
        count = 1
        while count <= N:
                total += count
                count += 1

        return total

Num = 7

result = sum_of_natural_numbers(Num)

print("The Sum of the First", Num, "Natural Numbers is:",          result)
```

PART A, PGM 4 – Display Multiplication Tables

```
num = 12
#num = int(input("enter a number"))
for i in range (1,11):
    print(num, 'x' , i ,"=", num*i)
```

PART A, PGM 5 – Check if a given number is a Prime Number or not

```
num = 5
# If given number is greater than 1
if num > 1:
   # Iterate from 2 to n // 2
   for i in range(2, num):
      if (num % i) == 0:
         print(num, "is not a prime number")
         break
   else:
      print(num, "is a prime number")
else:
   print(num, "is not a prime number")
```

PART A, PGM 6 – Implement a sequential search

```
# Function defination
def search(arr, N, x):

    for i in range(N):
        if (arr[i] == x):
            return i
    return -1

# Driver Code

arr = [2, 3, 4, 10, 40]
x = 4
N = len(arr)

# Function call
result = search(arr, N, x)
if(result == -1):
    print("Element is not present in array")
else:
    print("Element is present at index", result)
```

PART A, PGM 7 – Create a calculator program

```
def addition(a,b):
    sum=a+b
    return sum

def subtraction(a,b):
    diff=a-b
    return diff

def multiplication(a,b):
    pro=a*b
    return pro

def division(a,b):
    div=a/b
    return div

num1=int(input("enter the value of num1"))
num2=int(input("enter the value of num2"))

print("OPERATION")
print("1.Addition")
print("2.Subtraction")
print("3.Multiplication")
print("4.Division")
```

```
select=int(input("select a operation"))


if(select==1):
   print("the addition of",num1,"and",num2,"is",addition(num1,num2))
elif(select==2):
   print("the subtraction of",num1,"and",num2,"is",subtraction(num1,num2))
elif(select==3):
   print("the multiplication
of",num1,"and",num2,"is",multiplication(num1,num2))
elif(select==4):
   if num2==0:
      print("enter postive value for num2")
   else:
      print("the division of",num1,"and",num2,"is",division(num1,num2))
else:
   print("invalid input")
```

PART A, PGM 8 – Explore string functions

```
text = 'geeKs For geEkS'


print(text.upper())
print(text.lower())
print(text.title())
print(text.swapcase())
print(text.capitalize())
print(text)


word = 'find me if you can.'


char_count = word.count('n')
print(char_count)


print(word.find('me'))


result = word.endswith('you can')
print (result)
result = word.endswith('can.')
print (result)
result = word.endswith('you can.')
print (result)
result = word.endswith('find me if you can.')
print (result)
```

```
str1 = "HELLOWORLD"
str2 = "helloworld"

print(str1.islower())
print(str2.islower())
print(str1.isupper())
print(str2.isupper())

result = word.startswith('Find')
print (result)
result = word.startswith('find me')
print (result)
result = word.startswith('find me If')
print (result)
result = word.startswith('find me if you can.')
print (result)

replaced_text = str1.replace('HELLO', 'HEY')
print(replaced_text)
print(str2.replace('o', 'i'))
```

PART A, PGM 9 – Implement Selection Sort

```
def selectionsort(arr):
    n = len(arr)
    for i in range(n):
        mini = i
        for j in range(i+1,n):
            if arr[j]<arr[mini]:
                mini = j
        arr[i],arr[mini]=arr[mini],arr[i]#left , right = right , left (built-in method)


arr = [2, 33, 30, 5, 1]
selectionsort(arr)
print('The array after sorting in Ascending Order by selection sort is:',)
print(arr)
```

PART A, PGM 10 – Implement Stack

```
stack = [1,2,3] #stack is just the name of the list

print('Initial stack') #Initial stack
print(stack)

stack.append('a')
stack.append('b')

print('Updated stack') #Updated stack
print(stack)

print('Elements popped from stack:')
print(stack.pop())
print(stack.pop())

print('Stack after elements are popped:')
print(stack)
```

PART B, PGM 1 – Demonstrate usage of basic regular expression

```
import re
print("--MENU--")
print("1.Use of ^ caret")       #Starts with
print("2.Use of $ dollar")      #Ends with
print("3.Use of * asteric")     #Returns a list of occurrences
print("4.Use of \\ backslash")  #Signals a special sequence here,\d = special sequence to match digits
print("5.Exit")

while True:
    i=int(input("Enter Your Choice:"))
    s="The rain in Spain.37"
    if(i==1):
        x=re.findall("^The",s)
        if(x):
            print("Yes, the string-",s,"starts with The")
        else:
            print("No, the string-",s,"doesn't starts with The")
    elif i==2:
        x=re.findall("Spain$",s)
        if(x):
            print("Yes, the string-",s,"ends with Spain")
        else:
            print("No, the string-",s,"doesn't ends with Spain")
    elif i==3:
```

```
        x=re.findall("ai*",s)
        print(x)
    elif i==4:
        x=re.findall(r"\d",s) #raw string is used to make \d legal
        print(x)
    elif i==5:
        break
    else:
        print("Invalid Choice")
```

PART B, PGM 2 – Demonstrate use of advanced regular expressions for data validation

```
import re
str="The Rain in Spain"
while(True):
    print("")
    print("------------ADV REGULAR EXPRESSION--------------")
    print("1.Findall()")    #Returns a list containing all matches
    print("2.Search()")     #Returns a Match object if there is a match anywhere in the string
    print("3.Split()")     #Returns a list where the string has been split at each match
    print("4.Sub()")       #Replaces one or many matches with a string
    print("5.Exit")
    print("str=",str)

    ch=int(input("Enter Your Choice:"))
    if ch==1:
        x=re.findall("[a-m]",str)
        print("String Starts from a-m is:")
        print(x)
    elif ch==2:
        x=re.search("Rain",str)
        print("The 'Rain' word is located at position",x.start())
    elif ch==3:
        x=re.split(r"\s",str)   #"\s" Returns a match where the string contains a white space character
```

```
        print("Split String")
        print(x)
    elif ch==4:
        x=re.sub(r"\s","9",str) #raw string is used to make \s legal
        print("Replace all white space character with digit 9")
        print(x)
    elif ch==5:
        break
    else:
        print("Invalid Choice")
```

PART B, PGM 3 – Demonstrate use of List

```
list1 = [1, 5, 2, 9, 4, 5, 6, 3]

list2 = [8, 4, 7, 12, 14]


print(len(list1)) #print the length of the list i.e 8


list1.append(7) #adds at the end of the list (can add int, float, str)

print(list1)


list1.sort() #sorts in ascending order (works only with same data type)

print(list1)


list1.sort(reverse = True) #sorts in descending order (works only with same data
type)

print(list1)


sum = list1 + list2     #Adds list2 to list1

print(sum)


print(list1*2)          #repetition


print(5 in list1)


list1.insert(0, "apple") #inserts element at index 0 of any datatype
can insert any value

print(list1)
```

PART B, PGM 4 – Demonstrate use of Dictionaries

```
# Demonstration of Python Dictionaries

#Creating a dictionary

student = {"name": "Alice","age": 20,"course": "Computer Science","grades": [85, 90, 92]}


#Accessing values

print("Name:", student["name"])

print("Age:", student["age"])


#Adding a new key-value pair

student["email"] = "alice@example.com"

print("\nAfter adding email:")

print(student)


#Updating a value

student["age"] = 21

print("\nAfter updating age:")

print(student)


#Deleting a key-value pair

del student["course"]

print("\nAfter deleting course:")

print(student)
```

```
#Checking if a key exists
if "grades" in student:
    print("\ngrades exist in dictionary\n")

#Getting keys
print("Keys:", student.keys())

#Getting values
print("Values:", student.values())
```

PART B, PGM 5 – Create SQLite Database and Perform Operations on Tables

```
import sqlite3

conn=sqlite3.connect('example.db')

cursor=conn.cursor()

sql='''CREATE TABLE IF NOT EXISTS EMPLOY(EMPID CHAR(10) PRIMARY KEY, EMPNAME CHAR(20)
NOT NULL, DEPTNAME CHAR(20))'''

cursor.execute(sql)
print("employee table created successfully.....")

conn.commit()

print("Database Operation")
print("1.Insert into table")
print("2.Display contents from table")
print("3.Update contents of table")
print("4.Exit")

while True:
   ch=int(input("Enter Your Choice:"))
   if ch==1:
      eid=input("Enter EmpId:")
```

```
        ename=input("Enter EmpName:")
        dname=input("Enter DeptName:")
        sql='''INSERT INTO EMPLOY (EMPID, EMPNAME, DEPTNAME) VALUES
(?,?,?)'''
        vargs=(eid, ename, dname)
        cursor.execute(sql,vargs)
        print("Record Inserted Successfully...")
        conn.commit()

    elif ch==2:
        sql='''SELECT * FROM EMPLOY'''
        cursor.execute(sql)
        result=cursor.fetchall()
        print("EMPLOY Table has")
        for x in result:
            print(x)

    elif ch==3:
        eid=input("Enter EmpId to be updated:")
        newname=input("Enter updated name:")
        sql='''UPDATE EMPLOY SET EMPNAME= ? WHERE EMPID= ?'''
        vargs= (newname,eid)
        cursor.execute(sql,vargs)
        print("Record updated Successfully...")
        conn.commit()
    elif ch==4:
        conn.close()
```

```
        exit()
    else:
        print("Invalid Input")
```

PART B, PGM 6 – Create a GUI using Tkinter module

```
from tkinter import *

print("--Menu--")
print("1.Draw Rectangle")
print("2.Draw Oval")
print("3.Draw GUI controls-Label,Entry,Button")
print("4.Exit")

while True:
    i=int(input("Enter Your Choice:"))
    if i==1:
        top=Tk()
        c=Canvas(top,bg="blue",height=200,width=200)
        rec=c.create_rectangle(20,20,200,100,outline="red",fill="grey",width=2)
        c.pack()
        top.mainloop()

    elif i==2:
        top=Tk()
        c=Canvas(top,bg="blue",height=400,width=400)
        oval=c.create_oval(10,10,200,100,outline="red",fill="yellow",width=2)
        c.pack()
        top.mainloop()

    elif i==3:
```

```
    def show_entry_fields():
       print("First Name:",e1.get())
       print("Last Name:",e2.get())
    master=Tk()
    Label(master,text="First Name").grid(row=0)
    Label(master,text="Last Name").grid(row=1)
    e1=Entry(master)
    e2=Entry(master)
    e1.grid(row=0,column=1)
    e2.grid(row=1,column=1)

Button(master,text='Quit',command=quit).grid(row=3,column=0,sticky=W,pady
=4)



Button(master,text='Show',command=show_entry_fields).grid(row=3,column=
1,sticky=W,pady=4)
      break
  elif i==4:
      break
  else:
      print("Invalid Choice")
```

PART B, PGM 7 – Demonstrate Exceptions in Python

```
try:
    x = int(input("Please enter a number: "))
    result = 10 / x
    print("Result:", result)
except ValueError:
    print("Error: Invalid input. Please enter a valid number.")
except ZeroDivisionError:
    print("Error: Division by zero is not allowed.")
except Exception as e:
    print(f"An error occurred: {e} ")
finally:
    print("This finally block always executes, regardless of exceptions.")
```

PART B, PGM 8 – Drawing Line chart and Bar chart using Matplotlib

```
import matplotlib.pyplot as plt
print("MENU")
print("1.Line Chart")
print("2.Bar Chart")
ch=int(input("Enter Choice:"))

if ch==1:
    x=[20,10,40,2,45,67]
    y=[10,20,30,40,50,60]
    plt.figure(figsize=(8,6))
    plt.title("Line Plot Graph",fontsize=15,color='red')
    plt.xlabel("X Axis",fontsize=12,color='blue')
    plt.ylabel("Y Axis",fontsize=12,color='blue')
    plt.plot(x,y,color="purple",linewidth=5,label="Line Plot")
    plt.legend(loc=2,fontsize=12)
    plt.show()

elif ch==2:
    x=[20,10,30,40,2,45,67]
    y=[10,25,30,40,50,60,42]
    plt.figure(figsize=(8,6))
    plt.title("Bar Plot Graph",fontsize=15,color='red')
    plt.xlabel("X Axis",fontsize=12,color='blue')
    plt.ylabel("Y Axis",fontsize=12,color='blue')
```

```
    plt.bar(x,y,color="purple",linewidth=5,label="Bar Plot")
    plt.legend(loc=1,fontsize=12)
    plt.show()
else:
    print("Invalid Choice")
```

PART B, PGM 9 – Drawing Histogram and Pie chart using Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
print("MENU")
print("1.Histogram Chart:")
print("2.Pie Chart")
ch=int(input("Enter Choice:"))
if ch==1:
    np.random.seed(1)
    data=np.random.randint(1,100,50)
    plt.title("Histogram Plot",fontsize=15,color="red")
    plt.xlabel("X Axis",fontsize=12,color="blue")
    plt.ylabel("Y Axis",fontsize=12,color="blue")
    plt.hist(data)
    plt.show()
elif ch==2:
    data=[78,90,10,45,35]
    lab=["A","B","C","D","E"]
    col=["red","orange","blue","green","yellow"]
    plt.figure(figsize=(6,6))
    plt.title("Pie Plot",fontsize=15,color="red")
    plt.pie(data,colors=col,labels=lab,autopct="%.2f%%")
    plt.show()
else:
    print("Invalid Choice")
```

PART B, PGM 10 – Create Array using NumPy and Perform Operations on Array

```
import numpy as np


arr1 = np.array([5, 15, 20])

print('First array:')

print(arr1)


arr2 = np.array([2, 5, 9])

print('\nSecond array:')

print(arr2)



print("\nOPERATIONS:\n1.Addtion\n2.Subtraction\n3.Multiplication\n4.Divisio
n\n5.Power\n6.Remainder")


ch=int(input("Enter your choice:"))


if ch==1:
   print('\nAdding the two arrays:')
   print(np.add(arr1, arr2))
elif ch==2:
   print('\nSubtracting the two arrays:')
   print(np.subtract(arr1, arr2))
elif ch==3:
   print('\nMultiplying the two arrays:')
   print(np.multiply(arr1, arr2))
```

```
elif ch==4:
    print('\nDividing the two arrays:')
    print(np.divide(arr1, arr2))
elif ch==5:
    print('\nApplying power function again:')
    print(np.power(arr1, arr2))
elif ch==6:
    print('\nApplying remainder() function:')
    print(np.remainder(arr1, arr2))
else:
    print("Invalid Input")
```