1. **Write a Python class Account with attributes like account number, date of opening ,balance and customer name, and methods like deposit, withdraw, and check balance and account history.The class is from the view of banking site. Different objects has to be created for different account numbers.**

File: pythonq1.py

Code:

```
import datetime

class Account:

    def __init__(self, account_number, date_of_opening, balance, customer_name):
        self.account_number = account_number
        self.date_of_opening = date_of_opening
        self.balance = balance
        self.customer_name = customer_name
        self.account_history = []
        print("Account created for "+ self.customer_name+" with initial deposit "+str(self.balance)+" on 
"+self.date_of_opening)
        print("Account no for this customer is :"+self.account_number)

    def deposit(self, amount):
        self.balance += amount
        self.account_history.append("Deposit of $" + str(amount) + " on " + 
str(datetime.datetime.now()))
        print("Rs: "+str(amount)+" deposited on account no:"+self.account_number)

    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient funds")
        else:
            self.balance -= amount
            self.account_history.append("Withdrawal of $" + str(amount) + " on " + 
str(datetime.datetime.now()))
            print("Rs: "+str(amount)+"withdrawn from account no: "+self.account_number)

    def check_balance(self):
        print("The balance of account no: "+self.account_number+" is Rs: "+str(self.balance))

    def get_account_history(self):
        history=self.account_history
        print("Showing account history:---------")
        for h in history:
            print(h)



def Menu (acc):
    loop='Y'
    print("Bank account details of "+ str(acc.customer_name)+ " with account no 
"+str(acc.account_number))
    while(loop=='Y'):
        choice=input("Enter 1 for deposit,2 for withdrawn , 3 for balancecheck and 4 for accounthistory:: 
")
        if choice=='1':
            am1=int(input("Enter the amount to be deposited :: "))
            acc.deposit(am1)
        elif choice=='2':
            am1=int(input("Enter the amount to be withdrawn :: "))
            acc.withdraw(am1)
        elif choice=='3':
            acc.check_balance()
        elif choice=='4':
            acc.get_account_history()
        else:
            print("Bad Choice")

        loop=str(input("Do you want to continue Y/N:"))
        if(loop!='Y'):
            print("Ok ! The details of this user completed")



acc1 = Account("1234567890", "2023-07-23", 1000, "Suman Mukherjee")
Menu(acc1)
acc2 = Account("1234564589", "2023-07-21", 1000, "Souvik Das")
Menu(acc2)
```

Output:

```
Account created for Suman Mukherjee with initial deposit 1000 on 2023-07-23
Account no for this customer is :1234567890
Bank account details of Suman Mukherjee with account no 1234567890
Enter 1 for deposit,2 for withdrawn , 3 for balancecheck and 4 for accounthistory:: 1
Enter the amount to be deposited :: 2000
Rs: 2000 deposited on account no:1234567890
Do you want to continue Y/N:Y
Enter 1 for deposit,2 for withdrawn , 3 for balancecheck and 4 for accounthistory:: 3
The balance of account no: 1234567890 is Rs: 3000
Do you want to continue Y/N:N
Ok ! The details of this user completed
Account created for Souvik Das with initial deposit 1000 on 2023-07-21
Account no for this customer is :1234564589
Bank account details of Souvik Das with account no 1234564589
Enter 1 for deposit,2 for withdrawn , 3 for balancecheck and 4 for accounthistory:: 1
Enter the amount to be deposited :: 3000
Rs: 3000 deposited on account no:1234564589
Do you want to continue Y/N:Y
Enter 1 for deposit,2 for withdrawn , 3 for balancecheck and 4 for accounthistory:: 2
Enter the amount to be withdrawn :: 5000
Insufficient funds
Do you want to continue Y/N:N
Ok ! The details of this user completed
```

2. Write a python program that will generate a random integer list of length 24. Convert in into an numpy array. Reshape the array to a 6 × 4 matrix. Find transpose of the matrix.

File:pythonq2.py

Code:

```
import numpy as np

low = int(input("Enter lower limit of list:"))
high = int(input("Enter upper limit of list:"))

random_list = np.random.randint(low, high, 24)
print("The generated list of 26 numbers is: ")
print(random_list)

array = np.array(random_list)

matrix = array.reshape(6, 4)
print("Original matrix:")
print(matrix)

transpose = matrix.transpose()
print("Transpose of the matrix:")
print(transpose)
```

Output:

```
Enter lower limit of list:10
Enter upper limit of list:30
The generated list of 26 numbers is:
[14 21 26 19 21 14 12 15 19 17 17 13 27 18 27 16 28 29 23 15 10 13 14 26]
Original matrix:
[[14 21 26 19]
 [21 14 12 15]
 [19 17 17 13]
 [27 18 27 16]
 [28 29 23 15]
 [10 13 14 26]]
Transpose of the matrix:
[[14 21 19 27 28 10]
 [21 14 17 18 29 13]
 [26 12 17 27 23 14]
 [19 15 13 16 15 26]]
```

3. From internet download csv file of iris data. Read the csv file into a python program. Create a numpy matrix called data that will contain the numeric attributes of iris data. Create a list named target that will contain the species column of iris dataset.

File:pythonq3.py

Code:

```
import pandas as pd

import numpy as np

iris = pd.read_csv('Iris.csv')

#print("Reading data from Iris.csv File")

#print(iris)

#print("Type of iris: "+ str(type(iris)))

#print(iris.head)

data =np.array(list(iris.columns))

print("Printing numeric attributes of iris data: ")

print(data)

target=list(iris['Species'])

print("Printing species of iris dataset:")

print(target)
```

Output:

```
Printing numeric attributes of iris data:

['Id' 'SepalLengthCm' 'SepalWidthCm' 'PetalLengthCm' 'PetalWidthCm' 'Species']

Printing species of iris dataset:

['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-
setosa', 'Iris-setosa', 'Iris-setosa','Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa','Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-
setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa','Iris-setosa',
'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-
setosa', 'Iris-setosa','Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor','Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor','Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor','Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor','Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-versicolor','Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
'Iris-versicolor', 'Iris-virginica', 'Iris-virginica','Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-
virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica','Iris-virginica', 'Iris-virginica', 'Iris-virginica',
'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica','Iris-virginica', 'Iris-
virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-
virginica','Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
'Iris-virginica', 'Iris-virginica','Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-
virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica','Iris-virginica', 'Iris-virginica', 'Iris-virginica']
```

4. Write a recursive function in python to find the factorial of a number. Using the function find the sum of 1/n! where n =0 to n=10.

File:pythonq4.py

Code:

```
def factorial(n):
  if n == 0:
    return 1
  else:
    return n * factorial(n - 1)


def summation():
  sum = 0
  for n in range(11):
    #print(n)
    sum += 1 / factorial(n)
  return sum
print("The ans is:"+str(summation()))
```

Output:

```
The ans is:2.7182818011463845
```

5. Write a program that asks the user for a number and then prints out a list of all the divisors of that number.

File: pythonq5.py

Code:

```
def find_divisors(number):
  divisors = []
  for i in range(1, number + 1):
    if number % i == 0:
      divisors.append(i)
  return divisors


number = int(input("Enter a number: "))
divisors = find_divisors(number)
print("The divisors of {} are: {}".format(number, divisors))
```

Output:

```
  Enter a number: 18
The divisors of 18 are: [1, 2, 3, 6, 9, 18]
```

6. Write a program to create a list of prime numbers upto a number given as user input. (Use a user defined function to check prime number).

File:pythonq6.py

Code:

```
def is_prime(num):
    if num > 1:
        for i in range(2, int(num/2)+1):
            if (num % i) == 0:
                return False
        else:
            return True
```

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

primes = []

if (number == 1):

   print("1 is not prime no so it's not possible to return a list")

else:

   for i in range(2, number + 1):

      if is_prime(i):

         primes.append(i)

   print("The prime numbers upto given number is: "+str(primes))
```

Output:

```
Enter a number: 23

The prime numbers upto given number is: [2, 3, 5, 7, 11, 13, 17, 19, 23]
```

7. Write a program to find factorial of a user provided number using recursive function.

File:pythonq7.py

Code:

```
def factorial(n):

 if n == 0:

   return 1

 else:

   return n * factorial(n - 1)


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

print("The factorial of given number is: "+str(factorial(number)))
```

Output:

```
Enter a number: 5

The factorial of given number is: 120
```

8. Write a program in python to implement stack operation. Write your own function for push, pop and display.

File: pythonq8.py

Code:

```
def my_push(lst,value):
    lst.append(value)
    print(str(value)+" inserted in stack!")


def my_pop(lst):
    if len(lst)<1:
        print("Stack is empty")
    else:
        a=lst.pop()
        print("Poped element is: "+str(a))


def display(lst):
    i = len(lst)
    if len(lst)<1:
        print("Stack is empty")


    else:
        print("Stack elements are")
        for l in lst:
            print(l)




stack=[]
loop='Y'
while(loop=='Y'):
    choice=input("Enter 1 for push,2 for pop and 3 for display :: ")
    if choice=='1':
        val=int(input("Enter the value :: "))
        my_push(stack,val)
    elif choice=='2':
        my_pop(stack)
    elif choice=='3':
        display(stack)
    else:
        print("Bad Choice")

    loop=str(input("Do you want to continue Y/N:"))
    if(loop!='Y'):
        print("Ok ! Exited from this program")
```

Output:

```
Enter 1 for push,2 for pop and 3 for display :: 1
Enter the value :: 10
10 inserted in stack!
Do you want to continue Y/N:Y
Enter 1 for push,2 for pop and 3 for display :: 1
Enter the value :: 20
20 inserted in stack!
Do you want to continue Y/N:Y
Enter 1 for push,2 for pop and 3 for display :: 3
Stack elements are
10
20
Do you want to continue Y/N:Y
Enter 1 for push,2 for pop and 3 for display :: 2
```

```
Poped element is: 20

Do you want to continue Y/N:N

Ok ! Exited from this program
```

9. Write a program in python to implement queue operation. Write your own function for enqueue,dequeuq and display.

File:pythonq9.py

Code:

```
def enqueue(lst,value):
    lst.append(value)
    print(str(value)+" inserted in queue!")


def dequeue(lst):
    if len(lst)<1:
        print("Queue is empty")
    else:
        a=lst.pop(0)
        print("Poped element is: "+str(a))


def display(lst):
    i = len(lst)
    if len(lst)<1:
        print("Queue is empty")


    else:
        print("Queue elements are")
        for l in lst:
            print(l)
```

```
queue=[]
loop='Y'
while(loop=='Y'):
    choice=input("Enter 1 for enqueue,2 for dequeue and 3 for display :: ")
    if choice=='1':
        val=int(input("Enter the value :: "))
        enqueue(queue,val)
    elif choice=='2':
        dequeue(queue)
    elif choice=='3':
        display(queue)
    else:
        print("Bad Choice")

    loop=str(input("Do you want to continue Y/N:"))
    if(loop!='Y'):
        print("Ok ! Exited from this program")
```

Output:

```
Enter 1 for enqueue,2 for dequeue and 3 for display :: 1
Enter the value :: 10
10 inserted in queue!
Do you want to continue Y/N:Y
Enter 1 for enqueue,2 for dequeue and 3 for display :: 1
Enter the value :: 20
20 inserted in queue!
Do you want to continue Y/N:Y
Enter 1 for enqueue,2 for dequeue and 3 for display :: 3
Queue elements are
10
```

```
20

Do you want to continue Y/N:Y

Enter 1 for enqueue,2 for dequeue and 3 for display :: 2

Poped element is: 10

Do you want to continue Y/N:N

Ok ! Exited from this program
```