Database connectivity with Backend (Insert and Display Table records)

In [ ]:
```
import tkinter as tk
from mysql.connector import connect, Error
```

In [4]:
```
db_config = {
    'host':'127.0.0.1',
    'user':'root',
    'password':'manager',
    'database':'sys'
}
```

In [5]:
```
def execute_query(query, values=None):
    try:
        with connect(**db_config) as connection:
            cursor = connection.cursor()
            if values:
                cursor.execute(query, values)
            else:
                cursor.execute(query)
                connection.commit()
                return cursor.lastrowid

    except Error as e:
        print(f"Error executing query : {e}")
```

In [6]:
```
def insert_data():
    id = entry_id.get()
    name = entry_name.get()
    email = entry_email.get()
    query = "INSERT INTO student(id,name,email) VALUES (%s,%s,%s)"
    values = (id,name,email)
    execute_query(query, values)
```

In [9]:
```
def select_data():
    try:
        with connect(**db_config) as connection:
            query = "SELECT * FROM student"
            cursor = connection.cursor()
            cursor.execute(query)
            rows = cursor.fetchall()
            for row in rows:
                print(row)

    except Error as e:
        print(f"Error executing query : {e}")
```

In [ ]:
```
root = tk.Tk()
label_id = tk.Label(root, text = "ID")
label_id.grid(row = 0, column = 0)

label_id = tk.Entry(root)
label_id.grid(row = 0, column = 1)

label_name = tk.Label(root, text = "Name")
label_name.grid(row = 1, column = 0)

label_name = tk.Entry(root)
label_name.grid(row = 1, column = 1)

label_email = tk.Label(root, text = "Email")
label_email.grid(row = 2, column = 0)

label_email = tk.Entry(root)
label_email.grid(row = 2, column = 1)

button_insert = tk.Button(root, text = "Insert", command = insert_data)
button_insert.grid(row=4,column=0)

button_select = tk.Button(root, text = "Select", command = select_data)
button_select.grid(row=4,column=1)

root.mainloop()
```

Program to demonstrate GUI programming using Tkinter.

In [13]:
```
from tkinter import *

window = Tk()
window.geometry("500x300")
window.title("LOGIN")

l1 = Label(window,text="LOGIN",fg="WHITE",bg="BLACK",font="20")
l1.grid(row=1, column=2, pady=20)

l2 = Label(window, text="USERNAME : ")
l2.grid(row=2, column=1, padx=20, pady=20)

t2 = Entry(window, width = "50")
t2.grid(row=2,column=2, padx=50)

l3 = Label(window, text = "PASSWORD : ")
l3.grid(row=3, column=1, padx=2, pady=20)

t3 = Entry(window, width = "50")
t3.grid(row=3,column=2, padx=50)

l4 = Label(window, text = "PASSWORD SAVED : ")
l4.grid(row=4, column=1, padx=2, pady=20)

r1 = Radiobutton(window, text="YES")
r1.grid(row=4, column=2)
r2 = Radiobutton(window, text="NO")
r2.grid(row=5, column=2, padx="5", pady = 20)

window.mainloop()
```

Program for data structure algorithm using python for sorting

In [14]:
```
a = []
size = int(input("Enter the size of array : "))
for i in range(0,size):
    a.append(int(input()))
print("Before Sorted : ",a)

for i in range(0,len(a)):
    for j in range(0,len(a)-1):
        if(a[j]>a[j+1]):
            temp = a[j]
            a[j] = a[j+1]
            a[j+1] = temp

print("After Sorted : ",a)
```

```
Enter the size of array : 5
10
50
20
40
30
Before Sorted :  [10, 50, 20, 40, 30]
After Sorted :  [10, 20, 30, 40, 50]
```

Program to find out a factorial of given number.

In [5]:
```
num = int(input("Enter Number : "))
fact = 1
for i in range(1, num+1):
    fact = fact * i
print(fact)
```

```
Enter Number : 5
120
```

Python program for linear search

In [15]:
```
a = []
size = int(input("Enter the size of array : "))
for i in range(0,size):
    a.append(int(input()))

ele = int(input("Enter element for search : "))
f=0
for i in range(0,len(a)):
    if(ele == a[i]):
        f=1
        print("Element found at : ",i)
        break

if f==0:
    print("Element not found")
```

```
Enter the size of array : 5
1
3
4
2
5
Enter element for search : 2
Element found at :  3
```

Program to demonstrate class, object, Inheritance.

In [19]:
```
class first:
    a = "This is first class variable"
    def f_m(self):
        print("First Method ")

class second(first):
    def s_m(self):
        print("Second Method")

f = first()
f.f_m()
s = second()
s.f_m()
s.s_m()
```

```
First Method
First Method
Second Method
```

Use of DataFrame method and use of .csv files.

In [16]:
```
import pandas as pd
dataset = pd.read_csv("student_scores.csv")
print(dataset)
print(dataset.head(2))
print(dataset.tail(2))
print(dataset.info())
print(dataset.describe())
```

```
    Hours  Scores
0     2.5      21
1     5.1      47
2     3.2      27
3     8.5      75
4     3.5      30
5     1.5      20
6     9.2      88
7     5.5      60
8     8.3      81
9     2.7      25
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
15    8.9      95
16    2.5      30
17    1.9      24
18    6.1      67
19    7.4      69
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86
   Hours  Scores
0    2.5      21
1    5.1      47
    Hours  Scores
23    6.9      76
24    7.8      86
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 532.0 bytes
None
           Hours     Scores
count  25.000000  25.000000
mean    5.012000  51.480000
std     2.525094  25.286887
min     1.100000  17.000000
25%     2.700000  30.000000
50%     4.800000  47.000000
75%     7.400000  75.000000
max     9.200000  95.000000
```

Create a file, Write in to file, read a file, append the file.

In [22]:
```
wf = input("Enter the filename : ")
file = open(wf,'w')
file.write("File created")
file.close()

rf = input("Enter the filename : ")
file = open(rf,'r')
print(file.read())
file.close()

af = input("Enter the filename : ")
file = open(af,'a')
file.write("Content Updated")
file.close()
```

```
Enter the filename : rohit
Enter the filename : rohit
File created
Enter the filename : rohit
```

List and Dictionary with its important function (minimum 3)

In [24]:
```
l1 = [10,30,20,40]
print(l1)
l1.append([50])
print(l1)
l1.extend([90,80,60,70])
print(l1)
l1.insert(2,100)
print(l1)
print(l1.count(10))
l1.clear()
print(l1)

d1 = {"Name":"ROHIT BHOI", "S_NO" : "010101"}
print(d1.keys())
print(d1.values())
print(d1.items())
print(d1.popitem())
print(d1)
print(d1.pop("Name"))
d1.clear()
print(d1)
```

```
[10, 30, 20, 40]
[10, 30, 20, 40, [50]]
[10, 30, 20, 40, [50], 90, 80, 60, 70]
[10, 30, 100, 20, 40, [50], 90, 80, 60, 70]
1
[]
dict_keys(['Name', 'S_NO'])
dict_values(['ROHIT BHOI', '010101'])
dict_items([('Name', 'ROHIT BHOI'), ('S_NO', '010101')])
('S_NO', '010101')
{'Name': 'ROHIT BHOI'}
ROHIT BHOI
{}
```

Program to find out odd and even number up to given number.

In [25]:
```
num = int(input("Enter the Number : "))
even = []
odd = []
for i in range(1,num+1):
    if(i%2==0):
        even.append(i)
    else:
        odd.append(i)

print("Even : ",even)
print("Odd : ",odd)
```

```
Enter the Number : 20
Even :  [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Odd :  [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

Program to find out the given number is prime or not

In [17]:
```
num = int(input("Enter Number : "))
p = True
for i in range(2,num):
    if(num%i==0):
        p=False

if p==True:
    print("Number is Prime")
else:
    print("Number is not Prime")
```

```
Enter Number : 7
Number is Prime
```

Program to find out the given number is palindrome or not

In [34]:
```
num = int(input("Enter the Number : "))
temp = num
s=0
while(num>0):
    r=num%10
    s=s*10+r
    num=num//10

if s==temp:
    print("Number is Palindrome")
else:
    print("Number is not Palindrome")
```

```
Enter the Number : 554455
Number is Palindrome
```

Program to find out the given string is palindrome or not.

In [21]:
```
s1 = input("Enter a String : ")
rev = ""
for i in s1:
    rev=i+rev

if s1 == rev:
    print("String is palindrome")
else:
    print("String is not Palindrome")
```

```
Enter a String : nayan
String is palindrome
```

Program to find out the given number is Armstrong or not.

In [1]:
```
num = int(input("Enter the Number : "))
temp = num
s = 0
while(num>0):
    r=num%10
    s=s+(r*r*r)
    num=num//10

if s==temp:
    print("Number is Armstrong ")
else:
    print("Number is not Armstrong")
```

```
Enter the Number : 5252
Number is not Armstrong
```

Program to calculate the addition of odd number up to given range.

In [2]:
```
num = int(input("Enter the Number : "))
odd = []
for i in range(1,num+1):
    if(i%2 != 0):
        odd.append(i)

print("Addition : ",sum(odd))
```

```
Enter the Number : 5
Addition :  9
```

Program to calculate the addition of even number up to given range.

In [3]:
```
num = int(input("Enter the Number : "))
even = []
for i in range(1,num+1):
    if(i%2 == 0):
        even.append(i)

print("Addition : ",sum(even))
```

```
Enter the Number : 4
Addition :  6
```

Program to find out factorial using recursion.

In [4]:
```
def fact(num):
    if(num==1):
        return 1
    return num*fact(num-1)

num = int(input("Enter the Number : "))
print(fact(num))
```

```
Enter the Number : 5
120
```