1. INVOICE GENERATION – STATEMENT FLOW CONTROL

SOURCE CODE:

```
# Program 1 - INVOICE GENERATION
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: INVOICE GENERATION USING PYTHON")
print()
prods = int(input("How many products have you purchased? "))
print()
id_ls = []
name_ls = []
qty_ls = []
price_ls = []
dis_ls = []
discount = 0
total = 0
bill = 0
for loop in range(prods):
    pname = input(f"Enter the name of the product {loop+1}: ")
    pid = 'PDT-' + pname[0].upper() + str(loop)
    qty = int(input("Enter its quantity: "))
    price = float(input("Enter its price: "))
    if price >= 200 and price < 500:
        print("This product has a discount of 2 %")
        discount = price * 2/100
    elif price >= 500 and price < 1000:
        print("This product has a discount of 5 %")
        discount = price * 5/100
    elif price >= 1000 and price < 5000:
        print("This product has a discount of 8 %")
        discount = price * 8/100
    elif price >= 5000:
        print("This product has a discount of 10 %")
        discount = price * 10/100
    else:
        discount = 0
    name_ls.append(pname)
    id_ls.append(pid)
    qty_ls.append(qty)
```

```
    price_ls.append(price)
    dis_ls.append(discount)
    print()
print("\t\t\tFINAL INVOICE\n")
print("Prod_ID\t\tProd_Name\tQty\tPrice\tDiscount\tTotal")
for loop in range(prods):
    total = price_ls[loop] * qty_ls[loop] - dis_ls[loop]
    print(f"{id_ls[loop]}\t{name_ls[loop]}\t{qty_ls[loop]}\t{price_ls[loop]}\t{dis_ls[loop]}\t{total}")
    bill = bill + total
print(f"\n\t\t\t\tBILL AMOUNT:\t{bill}")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: INVOICE GENERATION USING PYTHON

How many products have you purchased? 5

Enter the name of the product 1: toffee
Enter its quantity: 10
Enter its price: 5

Enter the name of the product 2: soap
Enter its quantity: 7
Enter its price: 250
This product has a discount of 2 %

Enter the name of the product 3: lotion
Enter its quantity: 5
Enter its price: 800
This product has a discount of 5 %

Enter the name of the product 4: biscuit
Enter its quantity: 3
Enter its price: 1300
This product has a discount of 8 %

Enter the name of the product 5: chair
Enter its quantity: 1
Enter its price: 5200
This product has a discount of 10 %

                        FINAL INVOICE

Prod_ID         Prod_Name       Qty     Price   Discount        Total
PDT-T0          toffee          10      5.0     0               50.0
PDT-S1          soap            7       250.0   5.0             1745.0
PDT-L2          lotion          5       800.0   40.0            3960.0
PDT-B3          biscuit         3       1300.0  104.0           3796.0
PDT-C4          chair           1       5200.0  520.0           4680.0

                                BILL AMOUNT:    14231.0
```

RESULT:

The program is executed successfully.

2. SUM, PRODUCT, SQUARE OF ELEMENTS OF LIST

SOURCE CODE:

```
# Program 2 - SUM, PRODUCT, SQUARE OF ELEMENTS OF LIST
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: SUM, PRODUCT, SQUARE OF ELEMENTS OF LIST")
print()

Sum = 0
Product = 1
sqlst = []
elems = int(input("Enter the number of elements for the List: "))
List = []

for loop in range(elems):
    data = int(input(f"Enter element {loop+1}: "))
    List.append(data)
print("Your List -->", List)
print()
for e in List:
    Sum = Sum + e
    Product = Product * e
    Square = e * e
    sqlst.append(Square)
print("The Sum of the Elements of your List is:", Sum)
print("The Product of the Elements of your List is:", Product)
print("The Square of the Elements of your List is:", sqlst)
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: SUM, PRODUCT, SQUARE OF ELEMENTS OF LIST

Enter the number of elements for the List: 5
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Enter element 4: 4
Enter element 5: 5
Your List --> [1, 2, 3, 4, 5]

The Sum of the Elements of your List is: 15
The Product of the Elements of your List is: 120
The Square of the Elements of your List is: [1, 4, 9, 16, 25]
```

RESULT:

The program is executed successfully.

3. FIBONACCI SERIES

SOURCE CODE:

```
# Program 3 - FIBONACCI SERIES
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: FIBONACCI SERIES")
print()

def fibonacci(n):
    a = 0
    b = 1
    if n <= 0:
        print("Invalid Input !!!")
    elif n == 1:
        print("Fibonacci Series:\n", a)
    else:
        print("Fibonacci Series:\n", a, b, end=" ")
        for loop in range(n-2):
            c = a + b
            print(c, end=" ")
            a = b
            b = c
num = int(input("Enter the number of Fibonacci Series to be generated: "))
fibonacci(num)
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: FIBONACCI SERIES

Enter the number of Fibonacci Series to be generated: 13
Fibonacci Series:
 0 1 1 2 3 5 8 13 21 34 55 89 144
```

RESULT:

The program is executed successfully.

4. ATM OPERATIONS

SOURCE CODE:

```
# Program 4 - ATM OPERATIONS
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: ATM OPERATIONS")
print()
Balance = 130000
AcNo = int(input("Enter your Account Number: "))
Name = input("Enter your Name: ")
def Dep():
    global Balance
    print("\t\tDEPOSIT")
    print("\t\t-------")
    print("Your Existing Balance is:", Balance)
    depamt = int(input("Enter the amount to be deposited: "))
    Balance += depamt
    print(f"\n\tAmount of Rs.{depamt} is Credited Successfully")
    return Balance
def With():
    global Balance
    print("\t\tWITHDRAWAL")
    print("\t\t----------")
    print("Your Existing Balance is:", Balance)
    withamt = int(input("Enter the amount to be withdrawn: "))
    Balance -= withamt
    print(f"\n\tAmount of Rs.{withamt} is Debited Successfully")
    return Balance
def BalEnq():
    print("\n\t\tYour Account Number:", AcNo)
    print("\t\tYour Name:", Name)
    print("\n\t\tYour Account Balance is:", Balance)
print("\n\t\tWELCOME", Name)
print("\n\t1.Deposit")
print("\t2.Withdrawal")
print("\t3.Balance Enquiry")
print("\t4.Exit\n")
while True:
    choice = int(input("Choose your transaction: "))
```

```
    if choice == 1:
        print("\nAccount Number:", AcNo)
        print("Name:", Name)
        print("\n\tYour Account Balance is:",Dep())
    elif choice == 2:
        print("\nAccount Number:", AcNo)
        print("Name:", Name)
        print("\n\tYour Account Balance is:",With())
    elif choice == 3:
        BalEnq()
    elif choice == 4:
        break
    else:
        print("\n!!! CHOOSE A VALID TRANSACTION !!!")
    print()
print("\nThank You! visit again!")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: ATM OPERATIONS

Enter your Account Number: 130906
Enter your Name: Kathirvel

                WELCOME Kathirvel

        1.Deposit
        2.Withdrawal
        3.Balance Enquiry
        4.Exit

Choose your transaction: 1

Account Number: 130906
Name: Kathirvel
                DEPOSIT
                -------
Your Existing Balance is: 130000
Enter the amount to be deposited: 20000

        Amount of Rs.20000 is Credited Successfully

        Your Account Balance is: 150000
```

```
Choose your transaction: 2

Account Number: 130906
Name: Kathirvel
                WITHDRAWAL
                ----------
Your Existing Balance is: 150000
Enter the amount to be withdrawn: 80000

        Amount of Rs.80000 is Debited Successfully

        Your Account Balance is: 70000

Choose your transaction: 3

                Your Account Number: 130906
                Your Name: Kathirvel

                Your Account Balance is: 70000

Choose your transaction: 5

!!! CHOOSE A VALID TRANSACTION !!!

Choose your transaction: 4

Thank You! visit again!
```

RESULT:

The program is executed successfully.

5. AREA OF SHAPES

SOURCE CODE:

```
# Program 5 - AREA OF SHAPES
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: AREA OF SHAPES")
print()
print("\t\tAREA OF SHAPES")
print("\t\t--------------")
print("1. Area of Square")
print("2. Area of Rectangle")
print("3. Area of Triangle")
print("4. Area of Circle")
print("5. Exit\n")
choice = int(input("Enter your choice: "))
import math
def square(side):
    sqr = side * side
    print("The Area of Square:", sqr)
def rectangle(length, breadth):
    rect = length * breadth
    print("The Area of Rectangle:", rect)
def triangle(base, height):
    tri = 0.5 * base * height
    print("The Area of Triangle:", tri)
def circle(radius):
    circ = math.pi * radius ** 2
    print("The Area of Circle:", circ)
while True:
    if choice == 1:
        s = eval(input("Enter the value of Side: "))
        square(s)
    elif choice==2:
        l = eval(input("Enter the value of Length: "))
        b = eval(input("Enter the value of Breadth: "))
        rectangle(l,b)
    elif choice == 3:
        bse = eval(input("Enter the value of Base: "))
        h = eval(input("Enter the value of Height: "))
```

```
        triangle(bse,h)
    elif choice == 4:
        r = eval(input("Enter the value of Radius: "))
        circle(r)
    elif choice == 5:
        break
    else:
        print("INVALID SELECTION!!!")
    print()
    choice = int(input("Enter your choice: "))
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: AREA OF SHAPES

                AREA OF SHAPES
                --------------
1. Area of Square
2. Area of Rectangle
3. Area of Triangle
4. Area of Circle
5. Exit

Enter your choice: 1
Enter the value of Side: 5.2
The Area of Square: 27.040000000000003

Enter your choice: 2
Enter the value of Length: 20
Enter the value of Breadth: 30
The Area of Rectangle: 600

Enter your choice: 3
Enter the value of Base: 5.6
Enter the value of Height: 3.8
The Area of Triangle: 10.639999999999999

Enter your choice: 4
Enter the value of Radius: 25
The Area of Circle: 1963.4954084936207

Enter your choice: 7
INVALID SELECTION!!!

Enter your choice: 5
```

RESULT:

The program is executed successfully.

6. RANDOM GENERATOR (DICE GAME)

SOURCE CODE:

```
# Program 6 - RANDOM GENERATOR (DICE GAME)
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: RANDOM GENERATOR (DICE GAME)")
print()

import random
print("\t\tDICE GAME")
print("\t\t*********")
moves = int(input("How many moves do you want to play? "))

for loop in range(moves):
    print("\nMove", loop+1)
    print("\tYour Turn")
    print("\t---------")
    key = input("Press any key to roll the dice (Press Enter to skip)... ")
    if key:
        print("You -->", random.randint(1,6))
    else:
        print("Skipped...")
    print("\tComputer's Turn")
    print("\t---------------")
    print("Computer -->", random.randint(1,6))
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: RANDOM GENERATOR (DICE GAME)

                DICE GAME
                *********
How many moves do you want to play? 3

Move 1
        Your Turn
        ---------
Press any key to roll the dice (Press Enter to skip)... a
You --> 4
        Computer's Turn
        ---------------
Computer --> 6

Move 2
        Your Turn
        ---------
Press any key to roll the dice (Press Enter to skip)... 5
You --> 4
        Computer's Turn
        ---------------
Computer --> 4

Move 3
        Your Turn
        ---------
Press any key to roll the dice (Press Enter to skip)...
Skipped...
        Computer's Turn
        ---------------
Computer --> 1
```

RESULT:

The program is executed successfully.

7. COUNTING CHARACTERS, WORDS AND LINES

SOURCE CODE:

```
# Program 7 - COUNTING CHARACTERS, WORDS AND LINES
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: COUNTING CHARACTERS, WORDS AND LINES")
print()

file = open("Introduction_to_Python.txt","w+")
file.write('''Introduction to Python
1. Python is a high-level programming language known for its readability and simplicity.
2. Created by Guido van Rossum in the late 1980s, Python is continuously developed and maintained.
3. Python emphasizes clean and readable code, enhancing collaboration and ease of learning for
beginners.
4. It supports multiple programming paradigms, including procedural, object-oriented, and functional
programming.
5. Python boasts a comprehensive standard library, providing ready-to-use modules for various tasks.
6. It is widely used in web development, data science, artificial intelligence, and automation.
7. Python's large and active community contributes to its wealth of resources,
making it a preferred language for developers worldwide.''')

file.seek(0)
data = file.read()
num_of_char = len(data)
num_of_words = len(data.split())
num_of_lines = len(data.splitlines())
print("Number of characters is:", num_of_char)
print("Number of words is:", num_of_words)
print("Number of lines is:", num_of_lines)
file.close()
```

OUTPUT:

```
Introduction to Python
1. Python is a high-level programming language known for its readability and simplicity.
2. Created by Guido van Rossum in the late 1980s, Python is continuously developed and maintained.
3. Python emphasizes clean and readable code, enhancing collaboration and ease of learning for beginners.
4. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.
5. Python boasts a comprehensive standard library, providing ready-to-use modules for various tasks.
6. It is widely used in web development, data science, artificial intelligence, and automation.
7. Python's large and active community contributes to its wealth of resources,
making it a preferred language for developers worldwide.
```

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: COUNTING CHARACTERS, WORDS AND LINES

Number of characters is: 763
Number of words is: 106
Number of lines is: 9
```

RESULT:

The program is executed successfully.

8. REMOVING LINES CONTAINING 'a'

SOURCE CODE:

```
# Program 8 - REMOVING LINES CONTAINING 'a'
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: REMOVING LINES CONTAINING 'a'")
print()

rfile = open("Self_Discovery.txt","r+")
data = rfile.read()
print("The Original File Content:\n\n" + data)
print()
with open("Self_Discovery.txt","w+") as wfile:
    for line in data.splitlines():
        if 'a' not in line:
            wfile.write(line)
            wfile.write("\n")
    wfile.seek(0)
    content = wfile.read()
    print("The file content after removing lines containing 'a':\n\n" + content)
```

OUTPUT:

```
Rules for Self Discovery:
1. What we want most;
2. What we think about most;
3. How we use our money;
4. What we do with our leisure time;
5. The company we enjoy;
6. Who and what we admire;
7. What we laugh at.
```

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: REMOVING LINES CONTAINING 'a'

The Original File Content:

Rules for Self Discovery:
1. What we want most;
2. What we think about most;
3. How we use our money;
4. What we do with our leisure time;
5. The company we enjoy;
6. Who and what we admire;
7. What we laugh at.

The file content after removing lines containing 'a':

Rules for Self Discovery:
3. How we use our money;
```

```
1 Rules for Self Discovery:
2 3. How we use our money;
3
```

RESULT:

The program is executed successfully.

9. WORDS SEPARATOR BY '#'

SOURCE CODE:

```
# Program 9 - WORDS SEPARATOR BY '#'
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: WORDS SEPARATOR BY '#'")
print()

w_fle = open("Python.txt","w")
w_fle.write("'This is Data File Handling. Two Data Files are Text File and Binary File.'")
w_fle.flush()
with open("Python.txt","r") as r_fle:
    sent = r_fle.read()
    print("Original File Content:")
    print(sent)
    print("Words separated by '#':")
    for loop in range(len(sent)):
        if sent[loop].isspace():
            print('#', end='')
        else:
            print(sent[loop], end='')
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: WORDS SEPARATOR BY '#'

Original File Content:
This is Data File Handling. Two Data Files are Text File and Binary File.
Words separated by '#':
This#is#Data#File#Handling.#Two#Data#Files#are#Text#File#and#Binary#File.
```

```
1 This is Data File Handling. Two Data Files are Text File and Binary File.
```

RESULT:

The program is executed successfully.

10. SEARCHING IN A BINARY FILE

SOURCE CODE:

```
# Program 10 - SEARCHING IN A BINARY FILE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: SEARCHING IN A BINARY FILE")
print()

import pickle
students = []
num = int(input("How many records you want to enter: "))
for loop in range(num):
    rno = int(input(f"\nEnter the Roll Number of the Student {loop+1}: "))
    name = input(f"Enter the Name of the Student {loop+1}: ")
    cls = int(input(f"Enter the Class of the Student {loop+1}: "))
    rec = [rno,name,cls]
    students.append(rec)
with open("Students_Data.dat","wb+") as f:
    pickle.dump(students,f)
    f.seek(0)
    ls = pickle.load(f)
    search = int(input("\nEnter the Roll Number of the Student you want to search: "))
    for row in ls:
        if row[0] == search:
            print("\nRecord found...", row, sep="\n")
            break
    else:
        print("\nRecord Not Found")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: SEARCHING IN A BINARY FILE

How many records you want to enter: 3

Enter the Roll Number of the Student 1: 1
Enter the Name of the Student 1: ramesh
Enter the Class of the Student 1: 12

Enter the Roll Number of the Student 2: 2
Enter the Name of the Student 2: suresh
Enter the Class of the Student 2: 12

Enter the Roll Number of the Student 3: 3
Enter the Name of the Student 3: rajesh
Enter the Class of the Student 3: 12

Enter the Roll Number of the Student you want to search: 2

Record found...
[2, 'suresh', 12]
```

RESULT:

The program is executed successfully.

11. UPDATING BINARY FILE

SOURCE CODE:

```
# Program 11 - UPDATING BINARY FILE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: UPDATING BINARY FILE")
print()

import pickle
marklist = []
num = int(input("How many records you want to enter: "))
for loop in range(num):
    rno = int(input(f"\nEnter the Roll Number of the Student {loop+1}: "))
    name = input(f"Enter the Name of the Student {loop+1}: ")
    mark = float(input(f"Enter the Mark of the Student {loop+1}: "))
    det = [rno,name,mark]
    marklist.append(det)
with open("Mark_List.dat","wb+") as bfile:
    pickle.dump(marklist,bfile)
    bfile.seek(0)
    mlst = pickle.load(bfile)
    print("\nOld Mark List:", mlst, sep="\n")
    srh = int(input("\nEnter the Roll Number of the Student whose mark is to be updated: "))
    for rec in mlst:
        if rec[0] == srh:
            new = float(input(f"Enter the New Mark of the Student: "))
            rec[2] = new
            bfile.seek(0)
            pickle.dump(mlst,bfile)
            bfile.seek(0)
            updlst = pickle.load(bfile)
            print("Mark List Updated...")
            print(updlst)
            break
    else:
        print("Record Not Found...")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: UPDATING BINARY FILE

How many records you want to enter: 3

Enter the Roll Number of the Student 1: 1
Enter the Name of the Student 1: ramesh
Enter the Mark of the Student 1: 98

Enter the Roll Number of the Student 2: 2
Enter the Name of the Student 2: suresh
Enter the Mark of the Student 2: 73

Enter the Roll Number of the Student 3: 3
Enter the Name of the Student 3: rajesh
Enter the Mark of the Student 3: 36

Old Mark List:
[[1, 'ramesh', 98.0], [2, 'suresh', 73.0], [3, 'rajesh', 36.0]]

Enter the Roll Number of the Student whose mark is to be updated: 3
Enter the New Mark of the Student: 65.5
Mark List Updated...
[[1, 'ramesh', 98.0], [2, 'suresh', 73.0], [3, 'rajesh', 65.5]]
```

RESULT:

The program is executed successfully.

12. CREATING CSV FILE FOR STORING INVENTORY RECORDS

SOURCE CODE:

```
# Program 12- CREATING CSV FILE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: CREATING CSV FILE")
print()

import csv
invntry = {}
stkno = []
stkname = []
qty = []
choice = 'Y'
print("\t\tINVENTORY")
print("\t\t*********")
while choice.upper() == 'Y':
    sno = int(input("Enter the Stock Serial Number: "))
    name = input("Enter the Stock Name: ")
    q = int(input("Enter the Quantity: "))
    stkno.append(sno)
    stkname.append(name)
    qty.append(q)
    print()
    choice = input("Do you want to continue?(Y/N): ")
invntry.update(StockNo=stkno, StockName=stkname, Quantity=qty)

with open("Inventory.csv", "w+", newline="\n") as cfile:
    w = csv.writer(cfile)
    w.writerow(invntry.keys())
    for loop in range(len(stkno)):
        w.writerow([invntry['StockNo'][loop],invntry['StockName'][loop],invntry['Quantity'][loop]])
    r = csv.reader(cfile)
    cfile.seek(0)
    for row in r:
        print(row)
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: CREATING CSV FILE

                INVENTORY
                *********
Enter the Stock Serial Number: 1
Enter the Stock Name: pen
Enter the Quantity: 3

Do you want to continue?(Y/N): y
Enter the Stock Serial Number: 2
Enter the Stock Name: pencil
Enter the Quantity: 10

Do you want to continue?(Y/N): y
Enter the Stock Serial Number: 3
Enter the Stock Name: eraser
Enter the Quantity: 5

Do you want to continue?(Y/N): y
Enter the Stock Serial Number: 4
Enter the Stock Name: scale
Enter the Quantity: 2

Do you want to continue?(Y/N): n
['StockNo', 'StockName', 'Quantity']
['1', 'pen', '3']
['2', 'pencil', '10']
['3', 'eraser', '5']
['4', 'scale', '2']
```

	A	B	C
1	StockNo	StockName	Quantity
2	1	pen	3
3	2	pencil	10
4	3	eraser	5
5	4	scale	2
6			

RESULT:

The program is executed successfully.

13. 2D LIST (MATRIX) CREATION

SOURCE CODE:

```
# Program 13 - 2D LIST (MATRIX) CREATION
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: 2D LIST (MATRIX) CREATION")
print()
cols = int(input("Enter the number of columns you need: "))
rows = int(input("Enter the number of rows you need: "))
column = []
for r in range(rows):
    row = []
    for c in range(cols):
        rowelem = eval(input(f"Enter the value for row {r+1} and column {c+1}: "))
        row.append(rowelem)
    column.append(row)
    print()
print("Your 2D List:")
print(column)
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: 2D LIST (MATRIX) CREATION

Enter the number of columns you need: 3
Enter the number of rows you need: 4
Enter the value for row 1 and column 1: 'A'
Enter the value for row 1 and column 2: 'B'
Enter the value for row 1 and column 3: 'C'

Enter the value for row 2 and column 1: 'D'
Enter the value for row 2 and column 2: 'E'
Enter the value for row 2 and column 3: 'F'

Enter the value for row 3 and column 1: 'G'
Enter the value for row 3 and column 2: 'H'
Enter the value for row 3 and column 3: 'I'

Enter the value for row 4 and column 1: 'J'
Enter the value for row 4 and column 2: 'K'
Enter the value for row 4 and column 3: 'L'

Your 2D List:
[['A', 'B', 'C'], ['D', 'E', 'F'], ['G', 'H', 'I'], ['J', 'K', 'L']]
```

RESULT:

The program is executed successfully.

14. STACK OPERATIONS

SOURCE CODE:

```
# Program 14 - STACK OPERATIONS
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: STACK OPERATIONS")
print()
def Push(stk, elem):
    stk.append(elem)
    top = len(stk) - 1
def Pop(stk):
    if stk == []:
        return 'Underflow'
    else:
        del_elem = stk.pop()
        if len(stk) == 0:
            top = None
        else:
            top = len(stk) - 1
        return del_elem
def Peek(stk):
    if stk == []:
        return 'Underflow'
    else:
        top = len(stk) - 1
        return stk[top]
def Display(stk):
    if stk == []:
        print("\n\tStack is Empty !")
    else:
        top = len(stk) - 1
        print("\nStack contains:")
        print("\t", stk[top], "<- Top element")
        for loop in range(top-1, -1, -1):
            print("\t", stk[loop])
Stack = []
top = None
while True:
    print("\n\t\tSTACK OPERATIONS")
```

```
    print("\t\t***************")
    print("1.Push")
    print("2.Pop")
    print("3.Peek")
    print("4.Display")
    print("5.Exit")
    ch = int(input("Enter your choice: "))
    if ch < 1 or ch > 5:
        print("\n!!! INVALID SELECTION !!! Choose from 1 to 5")
    elif ch == 1:
        Element = int(input("Enter the element: "))
        Push(Stack, Element)
    elif ch == 2:
        d_elem = Pop(Stack)
        if d_elem == 'Underflow':
            print("\n\tUnderflow ! The Stack is Empty")
        else:
            print("\n\tThe Popped element is:", d_elem)
    elif ch == 3:
        top_elem = Peek(Stack)
        if top_elem == 'Underflow':
            print("\n\tUnderflow ! The Stack is Empty")
        else:
            print("\n\tThe Topmost element is:", top_elem)
    elif ch == 4:
        Display(Stack)
    elif ch == 5:
        break
print("Thank You!")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: STACK OPERATIONS


                STACK OPERATIONS
                ****************
1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter your choice: 1
Enter the element: 5

                STACK OPERATIONS
                ****************
1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter your choice: 1
Enter the element: 10

                STACK OPERATIONS
                ****************
 1.Push
 2.Pop
 3.Peek
 4.Display
 5.Exit
 Enter your choice: 4

 Stack contains:
          15 <- Top element
          10
          5

                STACK OPERATIONS
                ****************
 1.Push
 2.Pop
 3.Peek
 4.Display
 5.Exit
 Enter your choice: 2

         The Popped element is: 15

                STACK OPERATIONS
                ****************
 1.Push
 2.Pop
 3.Peek
 4.Display
 5.Exit
 Enter your choice: 3

         The Topmost element is: 10
```

```
                STACK OPERATIONS
                ****************
1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter your choice: 6

!!! INVALID SELECTION !!! Choose from 1 to 5

                STACK OPERATIONS
                ****************
1.Push
2.Pop
3.Peek
4.Display
5.Exit
Enter your choice: 5
Thank You!
```

RESULT:

The program is executed successfully.

15. LINEAR SEARCH

SOURCE CODE:

```
# Program 15 - LINEAR SEARCH
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: LINEAR SEARCH")
print()

def LinearSearch(lst, key):
    found = None
    for loop in range(len(lst)):
        if lst[loop] == key:
            found = lst.index(lst[loop])
    if found == None:
        return -1
    else:
        return found

ls = eval(input("Enter the list: "))
opt = 'Y'
if type(ls) == tuple:
    print("\t!!! ENTER THE ELEMENTS IN SQUARE BRACKETS[] !!!")
else:
    while opt == 'Y' or opt == 'y':
        search = int(input("\nEnter the element to be searched: "))
        ind = LinearSearch(ls,search)
        if ind == -1:
            print("\n\tSorry! Element is not in the List")
        else:
            print(f"\n\tThe Element {search} is found at Index Position {ind}")
        opt = input("\nDo you want to search any other elements? (Y/N): ")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: LINEAR SEARCH

Enter the list: [1,2,3,4,5]

Enter the element to be searched: 3

        The Element 3 is found at Index Position 2

Do you want to search any other elements? (Y/N): y

Enter the element to be searched: 6

        Sorry! Element is not in the List

Do you want to search any other elements? (Y/N): n
```

RESULT:

The program is executed successfully.

16. BINARY SEARCH

SOURCE CODE:

```
# Program 16 - BINARY SEARCH
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: BINARY SEARCH")
print()

def BinarySearch(lst,key):
    beg = 0
    last = len(ls) - 1
    while beg <= last:
        mid = (beg+last) // 2
        if search == ls[mid]:
            return mid
        elif search > ls[mid]:
            beg = mid+1
        elif search < ls[mid]:
            last = mid - 1
    else:
        return -1

ls = eval(input("Enter the list: "))
opt = 'y'
if type(ls) == tuple:
    print("\t!!! ENTER THE ELEMENTS IN SQUARE BRACKETS[] !!!")
else:
    while opt.lower() == 'y':
        ls.sort()
        print("The Sorted List:", ls)
        search = int(input("\nEnter the element to be searched: "))
        ind = BinarySearch(ls,search)
        if ind == -1:
            print("\n\tSorry! Element is not in the List")
        else:
            print(f"\n\tThe Element {search} is found at Index Position {ind}")
        opt = input("\nDo you want to search any other elements? (Y/N): ")
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: BINARY SEARCH

Enter the list: [15,10,30,25,5,20]
The Sorted List: [5, 10, 15, 20, 25, 30]

Enter the element to be searched: 25

        The Element 25 is found at Index Position 4

Do you want to search any other elements? (Y/N): y
The Sorted List: [5, 10, 15, 20, 25, 30]

Enter the element to be searched: 50

        Sorry! Element is not in the List

Do you want to search any other elements? (Y/N): n
```

RESULT:

The program is executed successfully.

17. GENERATING CREATE, INSERT QUERIES - PYTHON MYSQL INTERFACE

SOURCE CODE:

```
# Program 17 - GENERATING CREATE, INSERT QUERIES - PYTHON MYSQL INTERFACE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: GENERATING CREATE, INSERT QUERIES - PYTHON
MYSQL INTERFACE")
print()

import mysql.connector as sql
conn = sql.connect(host="localhost", user="root", password="kathir13", database="Class12CS")
if conn.is_connected():
    print("Connection Established Successfully... Process Started...")
crsr = conn.cursor()
create = '''CREATE TABLE Student
(RollNo VARCHAR(20), Name CHAR(20), DOB DATE, PhoneNo BIGINT, Location
VARCHAR(30))'''
crsr.execute(create)
conn.commit()
print("Student Table is Created Successfully...")
print("\tEnter Data for the Table Student:")
rec = int(input("How many records do you want to insert in the Table Student? "))
print()
for loop in range(rec):
    rno = input(f"Enter the Roll Number of the Student {loop+1}: ")
    name = input(f"Enter the Name of the Student {loop+1}: ")
    dob = int(input(f"Enter the DOB of the Student {loop+1} in YYYYMMDD format: "))
    phone = int(input(f"Enter the Phone Number of the Student {loop+1}: "))
    loc = input(f"Enter the Location of the Student {loop+1}: ")
    crsr.execute(f"INSERT INTO Student VALUES('{rno}','{name}',{dob},{phone},'{loc}')")
    conn.commit()
    print(f"\t{loop+1} Records Inserted Successfully...\n")
conn.close()
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: GENERATING CREATE, INSERT QUERIES - PYTHON MYSQL INTERFACE

Connection Established Successfully... Process Started...
Student Table is Created Successfully...
        Enter Data for the Table Student:
How many records do you want to insert in the Table Student? 3

Enter the Roll Number of the Student 1: 1
Enter the Name of the Student 1: Ramesh
Enter the DOB of the Student 1 in YYYYMMDD format: 20060522
Enter the Phone Number of the Student 1: 8956276492
Enter the Location of the Student 1: RS puram
        1 Records Inserted Successfully...

Enter the Roll Number of the Student 2: 2
Enter the Name of the Student 2: Suresh
Enter the DOB of the Student 2 in YYYYMMDD format: 20061127
Enter the Phone Number of the Student 2: 9871643251
Enter the Location of the Student 2: Gandhipuram
        2 Records Inserted Successfully...

Enter the Roll Number of the Student 3: 3
Enter the Name of the Student 3: Rajesh
Enter the DOB of the Student 3 in YYYYMMDD format: 20060113
Enter the Phone Number of the Student 3: 8713246985
Enter the Location of the Student 3: Peelamedu
        3 Records Inserted Successfully...
```

```
mysql> show tables;
+----------------------+
| Tables_in_class12cs  |
+----------------------+
| student              |
+----------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM student;
+--------+--------+------------+------------+-------------+
| RollNo | Name   | DOB        | PhoneNo    | Location    |
+--------+--------+------------+------------+-------------+
| 1      | Ramesh | 2006-05-22 | 8956276492 | RS puram    |
| 2      | Suresh | 2006-11-27 | 9871643251 | Gandhipuram |
| 3      | Rajesh | 2006-01-13 | 8713246985 | Peelamedu   |
+--------+--------+------------+------------+-------------+
3 rows in set (0.00 sec)

mysql> _
```

RESULT:

The program is executed successfully.

18. GENERATING UPDATE, DELETE QUERIES - PYTHON MYSQL INTERFACE

SOURCE CODE:

```
# Program 18 - GENERATING UPDATE, DELETE QUERIES - PYTHON MYSQL INTERFACE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: GENERATING UPDATE, DELETE QUERIES - PYTHON
MYSQL INTERFACE")
print()

import mysql.connector as sql
conn = sql.connect(host="localhost", user="root", password="kathir13", database="Class12CS")
if conn.is_connected():
    print("Connection Established Successfully... Process Started...\n")
crsr = conn.cursor()
print("\tUPDATE STUDENT PHONE NUMBER...")
ch = input("\nDo You Want To Update Phone No. of Any Record?(Y/N): ")
while ch == 'Y' or ch == 'y':
    name = input("\nEnter the Name of the Student: ")
    phone = int(input(f"Enter the New Phone Number of {name}: "))
    crsr.execute(f"UPDATE Student SET PhoneNo={phone} WHERE Name='{name}'")
    conn.commit()
    print(f"\n\t{name} Phone Number is Updated to {phone} Successfully...")
    ch = input("\nDo You Want To Update Phone No. of Any Other Record?(Y/N): ")

print("\n\tDELETE STUDENT RECORD...")
ch1 = input("\nDo You Want To Delete Any Record?(Y/N): ")
while ch1 == 'Y' or ch1 == 'y':
    rno=input("\nEnter the Roll Number of the Student: ")
    crsr.execute(f"DELETE FROM Student WHERE RollNo='{rno}'")
    conn.commit()
    print(f"\n\t{rno} Record is Deleted Successfully...")
    ch1 = input("\nDo You Want To Delete Any Other Record?(Y/N): ")
conn.close()
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: GENERATING UPDATE, DELETE QUERIES - PYTHON MYSQL INTERFACE

Connection Established Successfully... Process Started...

        UPDATE STUDENT PHONE NUMBER...

Do You Want To Update Phone No. of Any Record?(Y/N): y

Enter the Name of the Student: Ramesh
Enter the New Phone Number of Ramesh: 9834612575

        Ramesh Phone Number is Updated to 9834612575 Successfully...

Do You Want To Update Phone No. of Any Other Record?(Y/N): n

        DELETE STUDENT RECORD...

Do You Want To Delete Any Record?(Y/N): y

Enter the Roll Number of the Student: 3

        3 Record is Deleted Successfully...

Do You Want To Delete Any Other Record?(Y/N): n
```

```
mysql> SELECT * FROM student;
+--------+---------+------------+------------+-------------+
| RollNo | Name    | DOB        | PhoneNo    | Location    |
+--------+---------+------------+------------+-------------+
| 1      | Ramesh  | 2006-05-22 | 9834612575 | RS puram    |
| 2      | Suresh  | 2006-11-27 | 9871643251 | Gandhipuram |
+--------+---------+------------+------------+-------------+
2 rows in set (0.00 sec)
```

RESULT:

The program is executed successfully.

19. GENERATING SELECT QUERY - PYTHON MYSQL INTERFACE

SOURCE CODE:

```
# Program 19 - GENERATING SELECT QUERY - PYTHON MYSQL INTERFACE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: GENERATING SELECT QUERY - PYTHON MYSQL
INTERFACE")
print()

import mysql.connector as sql
conn = sql.connect(host="localhost", user="root", password="kathir13", database="Class12CS")
if conn.is_connected():
    print("Connection Established Successfully... Process Started...\n")
crsr = conn.cursor()
select = "SELECT * FROM Student ORDER BY DOB"
crsr.execute(select)
table = crsr.fetchall()
count = crsr.rowcount
print(f"The Number of Records retrieved are: {count}")
print("The Records in the Table Student:")
for row in table:
    print(row)

ch = input("\nDo You Want To Retrieve Any Specific Number Of Records?(Y/N): ")
while ch.upper() == 'Y':
    rec = int(input("\nHow many Records do you need? "))
    crsr.execute(select)
    data = crsr.fetchmany(rec)
    count2 = crsr.rowcount
    print(f"\nThe Number of Records now retrieved are: {count2}")
    print(f"{rec} Records from the Table Student:")
    for record in data:
        print(record)
    ch = 'N'
conn.close()
```

OUTPUT:

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: GENERATING SELECT QUERY - PYTHON MYSQL INTERFACE

Connection Established Successfully... Process Started...

The Number of Records retrieved are: 2
The Records in the Table Student:
('1', 'Ramesh', datetime.date(2006, 5, 22), 9834612575, 'RS puram')
('2', 'Suresh', datetime.date(2006, 11, 27), 9871643251, 'Gandhipuram')

Do You Want To Retrieve Any Specific Number Of Records?(Y/N): y

How many Records do you need? 1

The Number of Records now retrieved are: 1
1 Records from the Table Student:
('1', 'Ramesh', datetime.date(2006, 5, 22), 9834612575, 'RS puram')
```

RESULT:

The program is executed successfully.

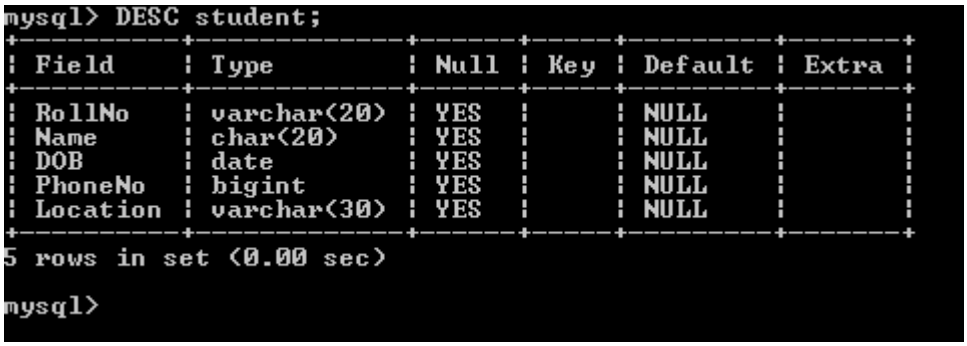
20. GENERATING ALTER QUERY - PYTHON SQL INTERFACE

SOURCE CODE:

```
# Program 20 - GENERATING ALTER QUERY - PYTHON SQL INTERFACE
print("NAME: Kathirvel")
print("ROLL NO: 12214")
print("CLASS: XII B")
print("NAME OF THE PROGRAM: GENERATING ALTER QUERY - PYTHON SQL
INTERFACE")
print()
import mysql.connector as sql
conn = sql.connect(host="localhost", user="root", password="kathir13", database="Class12CS")
if conn.is_connected():
   print("Connection Established Successfully... Process Started...")
crsr = conn.cursor()
print("\n\tAlter Operations for the Table Student")
print("\n1.Add New Column")
print("2.Modify Existing Column")
print("3.Rename Column")
print("4.Exit\n")
ch = int(input("Select The Operation You Want To Perform: "))
while True:
   if ch == 1:
      newcol = input("\nEnter the New Column Name: ")
      newdat = input(f"Enter the DataType with Size(if any) for {newcol}: ")
      crsr.execute(f"ALTER TABLE Student ADD {newcol} {newdat}")
      conn.commit()
      print("\n\tAdded a New Column Successfully...")
   elif ch == 2:
      col = input("\nEnter the Column Name to be Modified: ")
      chdat = input(f"Enter the New DataType with New Size(if any) for {col}: ")
      crsr.execute(f"ALTER TABLE Student MODIFY {col} {chdat}")
      conn.commit()
      print(f"\n\tModified {col} Successfully...")
   elif ch == 3:
      oldname = input("\nEnter the Column Name to be Renamed: ")
      newname = input(f"Enter the New Column Name with New Size(if any) for {oldname}: ")
      crsr.execute(f"ALTER TABLE Student CHANGE {oldname} {newname}")
      conn.commit()
      print(f"\n\tRenamed {oldname} Successfully...")
   elif ch == 4:
      break
```

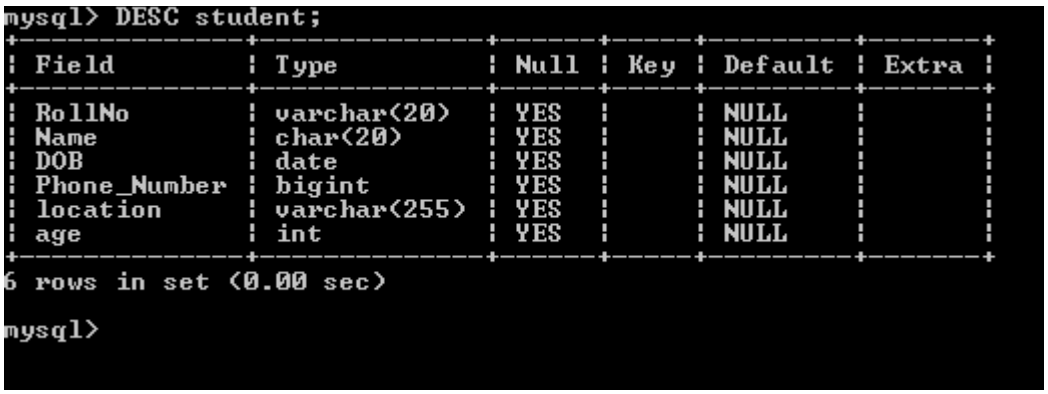
```
    else:
        print("!!! CHOOSE A VALID OPERATION !!!")
    ch = int(input("\nSelect The Operation You Want To Perform: "))
conn.close()
```

OUTPUT:



```
mysql> DESC student;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| RollNo   | varchar(20) | YES  |     | NULL    |       |
| Name     | char(20)    | YES  |     | NULL    |       |
| DOB      | date        | YES  |     | NULL    |       |
| PhoneNo  | bigint      | YES  |     | NULL    |       |
| Location | varchar(30) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

```
NAME: Kathirvel
ROLL NO: 12214
CLASS: XII B
NAME OF THE PROGRAM: GENERATING ALTER QUERY - PYTHON SQL INTERFACE

Connection Established Successfully... Process Started...

        Alter Operations for the Table Student

1.Add New Column
2.Modify Existing Column
3.Rename Column
4.Exit

Select The Operation You Want To Perform: 1

Enter the New Column Name: age
Enter the DataType with Size(if any) for age: int

        Added a New Column Successfully...

Select The Operation You Want To Perform: 2

Enter the Column Name to be Modified: location
Enter the New DataType with New Size(if any) for location: varchar(255)

        Modified location Successfully...

Select The Operation You Want To Perform: 3

Enter the Column Name to be Renamed: phoneno
Enter the New Column Name with New Size(if any) for phoneno: Phone_Number bigint

        Renamed phoneno Successfully...

Select The Operation You Want To Perform: 5
!!! CHOOSE A VALID OPERATION !!!

Select The Operation You Want To Perform: 4
```



```
mysql> DESC student;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| RollNo       | varchar(20)  | YES  |     | NULL    |       |
| Name         | char(20)     | YES  |     | NULL    |       |
| DOB          | date         | YES  |     | NULL    |       |
| Phone_Number | bigint       | YES  |     | NULL    |       |
| location     | varchar(255) | YES  |     | NULL    |       |
| age          | int          | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

RESULT:

The program is executed successfully.