1.Python Program to Make a Simple Calculator.

PROGRAM:

```
def add(x, y):
return x + y

def subtract(x, y):
return x - y

def multiply(x, y):
return x * y

def divide(x, y):
return x / y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
choice = input("Enter choice(1/2/3/4): ")
if choice in ('1', '2', '3', '4'):
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
if choice == '1':
print(num1, "+", num2, "=", add(num1, num2))
elif choice == '2':
print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':
print(num1, "*", num2, "=", multiply(num1, num2))
```

```
elif choice == '4':
print(num1, "/", num2, "=", divide(num1, num2))
break
else:
print("Invalid Input")
```

OUTPUT:

```
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 1
Enter first number: 23
Enter second number: 45
23.0 + 45.0 = 68.0
```

RESULT:

Thus the program has been verified and completed successfully.

2. Factorial of a number using recursion

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


num = 5
# check if the number is negative
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

Output:

```
The Factorial of 5 is 120
```

3. Python Program to Transpose a Matrix

AIM:

TO WRITE A Python Program for Transposing a Matrix

PROGRAM:

```
X = [[12,7],
    [4 ,5],
    [3 ,8]]

result = [[0,0,0],
         [0,0,0]]

# iterate through rows
for i in range(len(X)):
   # iterate through columns
   for j in range(len(X[0])):
       result[j][i] = X[i][j]

for r in result:
   print(r)
```

OUTPUT:

```
[12, 4, 3]
[7, 5, 8]
```

RESULT:

Thus the program has been verified and completed successfully.

4.Python Program to Multiply Two Matrices

To write a Python Program for Multiplying Two Matrices.

PROGRAM:

```
X = [[12,7,3],
    [4 ,5,6],
    [7 ,8,9]]
# 3x4 matrix
Y = [[5,8,1,2],
    [6,7,3,0],
    [4,5,9,1]]
# result is 3x4
result = [[0,0,0,0],
          [0,0,0,0],
          [0,0,0,0]]

for i in range(len(X)):
   for j in range(len(Y[0])):
       for k in range(len(Y)):
           result[i][j] += X[i][k] * Y[k][j]
for r in result:
   print(r)
```

OUTPUT:

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

RESULT:

Thus the program has been verified and completed successfully.

5.Python Program for Classes,Constructors and Objects

```
class Dog:

    animal = 'dog'

    def __init__(self, breed, color):

        self.breed = breed
        self.color = color

Rodger = Dog("Pug", "brown")
Buzo = Dog("Bulldog", "black")

print('Rodger details:')
print('Rodger is a', Rodger.animal)
print('Breed: ', Rodger.breed)
print('Color: ', Rodger.color)

print('\nBuzo details:')
print('Buzo is a', Buzo.animal)
print('Breed: ', Buzo.breed)
print('Color: ', Buzo.color)

print("\nAccessing class variable using class name")
print(Dog.animal)
```

Output:

```
Rodger details:

Rodger is a dog

Breed:  Pug

Color:  brown


Buzo details:

Buzo is a dog

Breed:  Bulldog

Color:  black


Accessing class variable using class name dog
```

6.Python Program for Dictionary

```
thisdict = {
  "name": "Rakesh",
  "Native": "Chennai",
  "year": 2001
}
#thisdict
print("1.print the dictionary:\n",thisdict)
#len()
print("\n2.return the number of entries in thisdict:",len(thisdict))
#keys()
print("\n3.Return the list of the keys",thisdict.keys())
#values()
print("\n4.Return the list of the value",thisdict.values())
#items
print("\n5.Return the list of the tuples containing keys and values",thisdict.items())
#copy()
print("\n6.Returns a copy of the dictionary",thisdict.copy())
#pop()
print("\n7.Removes the element with the specified key:",thisdict.pop("year"))
#get()
print("\n8.Removes the element with the specified key:",thisdict.get("name"))
#setdefault
print("\n9.Returns the value of the specified key:",thisdict.setdefault("name"))
```

```
#popitem

print("\n10.The removed item is the return value of the popitem() method, as a
tuple:",thisdict.popitem())

#update

thisdict.update({"color": "Black"})

print("\n11.",thisdict)

#clear

print("\n12.Remove all keys",thisdict.clear())
```

Output:

1.print the dictionary:

{'name': 'Rakesh', 'Native': 'Chennai', 'year': 2001}

2.return the number of entries in thisdict: 3

3.Return the list of the keys dict_keys(['name', 'Native', 'year'])

4.Return the list of the value dict_values(['Rakesh', 'Chennai', 2001])

5.Return the list of the tuples containing keys and valuesdict_items([('name', 'Rakesh'), ('Native', 'Chennai'), ('year', 2001)])

6.Returns a copy of the dictionary {'name': 'Rakesh', 'Native': 'Chennai', 'year': 2001}

7.Removes the element with the specified key: 2001

8.Removes the element with the specified key: Rakesh

9.Returns the value of the specified key: Rakesh

10.The removed item is the return value of the popitem() method, as a tuple: ('Native', 'Chennai')

11. {'name': 'Rakesh', 'color': 'Black'}

12.Remove all keys None

7.Python Program for List

```
n = int(input("Enter the size of the list "))
print("\n")
num_list = list(int(num) for num in input("Enter the list items ").strip().split())[:n]
print("User list: ", num_list)
print("negative indexing:",num_list[-1])
print("list slicing: ", num_list[slice(4)])
num_list.reverse()
print("Reversed string: ", num_list)
print("length of List: ",len(num_list))
num_list.insert(9,11)
print("insertion List: ",num_list)
num_list.append(7)
print("Append List: ",num_list)
print("concatenate  list",num_list + [14,12,13])
print("Pop List",num_list.pop())
num_list.remove(5)
print("remove list: ",num_list)
```

Output:

Enter the size of the list 5

Enter the list items 1 2 3 4 5

User list: [1, 2, 3, 4, 5]

negative indexing: 5

list slicing: [1, 2, 3, 4]

Reversed string: [5, 4, 3, 2, 1]

length of List: 5

insertion List: [5, 4, 3, 2, 1, 11]

Append List: [5, 4, 3, 2, 1, 11, 7]

concatenate list [5, 4, 3, 2, 1, 11, 7, 14, 12, 13]

Pop List 7

remove list: [4, 3, 2, 1, 11]

>

8.Python Program for Tuple

```
# Empty tuple
my_tuple = ()
print(my_tuple)

# Tuple having integers
my_tuple = (1, 2, 3)
print(my_tuple)

# tuple with mixed datatypes
my_tuple = (1, "Hello", 3.4)
print(my_tuple)

# nested tuple
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
```

Output

```
()
(1, 2, 3)
(1, 'Hello', 3.4)
('mouse', [8, 4, 6], (1, 2, 3))
```