Program Name: demo1.py

Write and execute the program

```
def display():
        print("welcome to function concept")


display()
```

output

Program Name: demo2.py

Write and execute the program

```
def display&()
        print("welcome to function concept")


display&()
```

output

Program Name: demo3.py

```
def display123():
print("welcome to function concept")


display123()
```

output

Program Name: demo4.py

```
def display(a):
        print("welcome to function concept", a)


display(10)
```

output

Program Name: Write and execute the program
demo5.py

```
def display(a):
        print("welcome to function concept", a)


display(99.7)
```

output

Program Name: Write and execute the program
demo6.py

```
def display(a):
        print("welcome to function concept", a)


display("Daniel")
```

output

Program Name: demo7.py

```
def display(a):
    for i in a:
        print(i)

display("Daniel")
```

output

Program Name: demo8.py

```
def display(a):
    print("welcome to function concept", a)

display([10, 20, 30, 40])
```

output

Program Name: demo9.py

Write and execute the program

```
def display(a):
        print("welcome to function concept", a)


b = [10, 20, 30, 40]
display(b)
```

output

Program Name: demo10.py

Write and execute the program

```
def display(a):
        for i in a:
                print(i)

b = [10, 20, 30, 40]
display(b)
```

output

Program Name: demo11.py

```
def display(a):
        for i in a:
                print(i+2)

b = [10, 20, 30, 40]
display(b)
```

output

Program Name: demo12.py

```
def display(a):
        print("welcome to function concept", a)


b = [10, "Daniel", 33.5]
display(b)
```

output

Program Name: demo13.py

```
def display(a):
        for i in a:
                print(i)

b = [10, "Daniel", 33.5]
display(b)
```

output

Program Name: demo14.py

```
def display(a):
        for i in a:
                print(i+1)

b = [10, "Daniel", 33.5]
display(b)
```

output

Program Name: Write and execute the program demo15.py

```
def display(a):
        print("welcome to function concept", a)


b = (10, 20, 30, 40)
display(b)
```

output

Program Name: Write and execute the program demo16.py

```
def display(a):
        for i in a:
                print(i)

b = (10, 20, 30, 40)
display(b)
```

output

Program Name: Write and execute the program
demo17.py

```
def display(a):
        print("welcome to function concept", a)


b = {10, 20, 30, 40}
display(b)
```

output

Program Name: Write and execute the program
demo18.py

```
def display(a):
        for i in a:
                print(i)

b = {10, 20, 30, 40}
display(b)
```

output

Program Name: demo19.py

```
def display(a):
        print("welcome to function concept", a)


b = {"id": 101, "name": "Daniel"}
display(b)
```

output

Program Name: demo20.py

```
def display(a):
        for i in a:
                print(i)

b = {"id": 101, "name": "Daniel"}
display(b)
```

output

Program Name: Write and execute the program demo21.py

```
def display(a):
        for i in a:
                print(i, a[i])

b = {"id": 101, "name": "Daniel"}
display(b)
```

output

Program Name: Please execute the program and check the output demo22.py

```
def m1():
        a = 33
        b = 200
        if b > a:
                print("b is greater than a")

m1()
```

Output

Program Name: Write and execute the program demo23.py

```
def m1(a, b):
        if b > a:
                print("b is greater than a")

m1(33, 200)
```

Output

Program Name: Write and execute the program demo24.py

```
def m1():
        a = 33
        b = 33

        if b > a:
                print("b is greater than a")

        elif a == b:
                print("a and b are equal")

m1()
```

Output

Program Name: Write and execute the program demo25.py

```
def m1(a, b):
    if b > a:
        print("b is greater than a")

    elif a == b:
        print("a and b are equal")

m1(33, 33)
```

Output

Program Name: Function to find the Max of three numbers
demo26.py

```
def m1():
    a = int(input('Enter first number  : '))
    b = int(input('Enter second number : '))
    c = int(input('Enter third number  : '))

    largest = 0

    if a > b and a > c:
        largest = a
    elif b > a and b > c:
        largest = b
    elif c > a and c > b:
        largest = c

    print(largest, "is the largest of three numbers.")

m1()
```

Output

Program Name: Function to find the Max of three numbers
demo27.py

```
def m1(p):
        result = max(p)
        print(result, "is the largest of three numbers.")

a = [10, 20, 30]
m1(a)
```

Output

Program Name: Function to sum all values
demo28.py

```
def addition(numbers):
        total = 0
        for x in numbers:
                total = total + x
        return total

result = addition([8, 2, 3, 0, 7])

print(result)
```

Output

Program Name: Function to multiply all values
demo29.py

```
def multiply(numbers):
    total = 1
    for x in numbers:
        total = total * x
    return total

result = multiply([1, 2, 3, 4])
print(result)
```

Output

Program Name: Check whether a number falls in a given range
demo30.py

```
def test_range(n):
    if n in range(3, 9):
        print( "The value is within the range", n)
    else:
        print("The number is outside the given range.")

test_range(5)
```

Output

Program Name: Calculate the number of upper / lower case letters in a string
demo31.py

```
text = input("Enter a string:")
count1 = 0
count2 = 0

for i in text:
        if i.islower():
                count1 = count1+1
        elif i.isupper():
                count2 = count2+1

print("The number of lowercase characters is:", count1)
print("The number of uppercase characters is:", count2)
```

Output

Program Name: Unique elements from a list
demo32.py

```
def unique_list(p):
        x = []
        for a in p:
                if a not in x:
                        x.append(a)
        return x

result = unique_list([1, 2, 3, 3, 3, 3, 4, 5])
print(result)
```

Output

Program Name: Write and execute the program demo33.py

```
def is_even_num(a):
        enum = []
        for i in a:
                if i % 2 == 0:
                        enum.append(i)
        return enum

b = [1, 2, 3, 4, 5, 6, 7, 8, 9]
result = is_even_num(b)
print(result)
```

Output

Program Name: Write and execute the program demo34.py

```
def display():
        a = list()
        r = range(1, 10)
        for i in r:
                a.append(i**2)
        print(a)

display()
```

Output

Program Name: Write and execute the program demo35.py

```
def display():
    a = []
    r = range(1, 10)
    for i in r:
        a.append(i**2)
    print(a)

display()
```

Output

Program Name: Write and execute the program demo36.py

```
from time import sleep

def traffic(r):
    for i in r:
        sleep(2)
        print(i)

traffic(range(1, 11))
```

Output

Program Name: Write and execute the program demo37.py

```
def 123display():
        print("welcome to function concept")


123display()
```

output

Program Name: Write and execute the program demo38.py

```
def display123():
        print("welcome to function concept")

display123()
```

output

Program Name: demo39.py

```
def display_one():
    print("welcome to function concept")

display_one()
```

output

Program Name: demo40.py

```
def testing(a, b):
    print("two parameterised function:", a, b)

testing(10, 20)
```

output

Program Name: demo41.py

```
def testing(a, a):
        print("two parameterised function:", a, a)

testing(10, 20)
```

output

Program Name: demo42.py

```
def wish():
        print("Hello")
        print("How are you")
        return 100

b = wish()
print(b)
```

output

Program Name: Write and execute the program demo43.py

```
def wish():
        print("Hello")
        return 100
        print("How are you")

b = wish()
print(b)
```

output

Program Name: Write and execute the program demo44.py

```
def wish():
        print("Hello")
        return 100
        return 111

b = wish()
print(b)
```

output

Program Name: Write and execute the program demo45.py

```
def balance():
    print("My bank balance is: ")
    return 100

print(balance())
```

output

Program Name: Write and execute the program demo46.py

```
def details():
    id = 101
    name = "Daniel"
    salary = 10000
    return id, name, salary

result = details()
print("all values:", result)
```

output

Program Name: Write and execute the program demo47.py

```
def sub(x, y):
        print(x-y)

sub(20, 10, 30)
```

output

Program Name: Write and execute the program demo48.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart("bangles", 20000)
```

output

Program Name: Write and execute the program demo49.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart(product = "bangles", 20000)
```

output

Program Name: Write and execute the program demo50.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart("bangles", price = 20000)
```

output

Program Name: Write and execute the program demo51.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart(product = "bangles", price = 20000)
```

output

Program Name: Write and execute the program demo52.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart(prod = "bangles", price = 20000)
```

output

Program Name: Write and execute the program demo53.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart(product = "bangles", pri = 20000)
```

output

Program Name: Write and execute the program demo54.py

```
def cart(product, price):
        print("Product is :" , product)
        print("cost is :" , price)

cart(prod = "bangles", pri = 20000)
```

output

Program Name: Write and execute the program demo55.py

```
def cart(product, price = 40.0):
        print("Product is :" , product)
        print("cost is :" , price)

cart(product = "pen")
```

output

Program Name: Write and execute the program demo56.py

```
def cart(product, price = 40.0):
        print("product is :", product)
        print("cost is :", price)

cart(product = "handbag", price = 10000)
```

output

Program Name: Write and execute the program demo57.py

```
def cart(product = "handbag", price):
        print("product is :", product)
        print("cost is :", price)

cart(price = 10000)
```

output

Program Name: Write and execute the program demo58.py

```
def m(x):
        print(x)

m(10)
```

output

Program Name: Write and execute the program demo59.py

```
def m(x):
    print(x)

m(10, 20)
```

output

Program Name: Write and execute the program demo60.py

```
def m(*x):
    print(x)

m(10)
```

output

Program Name: Write and execute the program demo61.py

```
def m(*x):
    print(x)

m(10, 20)
```

output

Program Name: Write and execute the program demo62.py

```
def m(*x):
        print(x)

m(10, 20, 30)
```

output

Program Name: Write and execute the program demo63.py

```
def m(a, *x):
        print(a)
        print(x)

m(10, 20, 30)
```

output

Program Name: demo64.py

```
def m(*x, a):
        print(a)
        print(x)

m(10, 20, 30)
```

output

Program Name: demo65.py

```
def display(**kwargs):
        print(kwargs)

display(id = 1, name = "Daniel", qualification = "MCA")
```

Output

Program Name: Write and execute the program demo66.py

```
a = lambda b : b + 15
result = a(10)

print(result)
print(a(10))
```

output

Program Name: Write and execute the program demo67.py

```
a = lambda b : b * 15
result = a(10)

print(result)
print(a(10))
```

output

Program Name: Write and execute the program demo68.py

```
a = lambda x, y : x * y
result = a(10, 20)

print(result)
print(a(10, 20))
```

output

Program Name: Write and execute the program demo69.py

```
def func_compute(n):
        return lambda x : x * n

result = func_compute(2)
print("Double the number of 15 =", result(15))

result = func_compute(3)
print("Triple the number of 15 =", result(15))

result = func_compute(4)
print("Quadruple the number of 15 =", result(15))

result = func_compute(5)
print("Quintuple the number 15 =", result(15))
```

output

Program Name: Sort a list of tuples using Lambda.

demo70.py

```
subject_marks = [('English', 88), ('Science', 90), ('Maths', 97),
('Social sciences', 82)]

print("Original list of tuples:")
print(subject_marks)

subject_marks.sort(key = lambda x: x[1])

print("Sorting the List of Tuples:")
print(subject_marks)
```

output

Program Name: Sort a list of dictionaries using Lambda
demo71.py

```
models = [{'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make':
'Mi Max', 'model':'2', 'color': 'Gold'}, {'make': 'Samsung', 'model':
7, 'color': 'Blue'}]

print("Original list of dictionaries :")
print(models)

sorted_models = sorted(models, key = lambda x: x['color'])

print("Sorting the List of dictionaries :")
print(sorted_models)
```

output

Program Name: Filter a list of integers using Lambda

demo72.py

```
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print("Original list of integers:")
print(nums)

print("Even numbers from the said list:")
even_nums = list(filter(lambda x: x%2 == 0, nums))
print(even_nums)

print("Odd numbers from the said list:")
odd_nums = list(filter(lambda x: x%2 != 0, nums))
print(odd_nums)
```

output

Program Name: square and cube every number in a given list using Lambda.
demo73.py

```
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print("Original list of integers:")
print(nums)

print("Square every number of the said list:")
square_nums = list(map(lambda x: x ** 2, nums))
print(square_nums)

print("Cube every number of the said list:")
cube_nums = list(map(lambda x: x ** 3, nums))
print(cube_nums)
```

output

Program Name: Find if a given string starts with a given character using Lambda
demo74.py

```
starts_with = lambda x: True if x.startswith('P') else False

print(starts_with('Python'))
print(starts_with('Java'))
```

output

Program Name: Extract year, month, date and time using Lambda
demo75.py

```
import datetime

now = datetime.datetime.now()

print(now)

year = lambda x: x.year
month = lambda x: x.month
day = lambda x: x.day
t = lambda x: x.time()

print(year(now))
print(month(now))
print(day(now))
print(t(now))
```

output

Program Name: Intersection of two given arrays using Lambda
demo76.py

```
array_nums1 = [1, 2, 3, 5, 7, 8, 9, 10]
array_nums2 = [1, 2, 4, 8, 9]

print("Original arrays:")
print(array_nums1)
print(array_nums2)

result = list(filter(lambda x: x in array_nums1, array_nums2))
print ("Intersection of the said arrays: ",result)
```

output

Program Name: positive and negative numbers in a given array using Lambda
demo77.py

```
array_nums = [-1, 2, -3, 5, 7, 8, 9, -10]
print("Original arrays:")
print(array_nums)

result = sorted(array_nums, key = lambda i: 0 if i == 0 else -1 / i)
print("Rearrange positive and negative numbers of the said
array:")

print(result)
```

output

Program Name: even and odd numbers in a given integers using Lambda
demo78.py

```
array_nums = [1, 2, 3, 5, 7, 8, 9, 10]
print("Original arrays:")
print(array_nums)

odd_ctr = len(list(filter(lambda x: (x%2 != 0) , array_nums)))
even_ctr = len(list(filter(lambda x: (x%2 == 0) , array_nums)))

print("Number of even numbers in the above array: ", even_ctr)
print("Number of odd numbers in the above array: ", odd_ctr)
```

output

Program Name: If the values in the list have a length of 6 using Lambda
demo79.py

```
weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday', 'Saturday', 'Sunday']

days = filter(lambda day: day if len(day)==6 else '', weekdays)

for d in days:
        print(d)
```

output

Program Name: add two given lists using map and lambda
demo80.py

```
nums1 = [1, 2, 3]
nums2 = [4, 5, 6]

print("Original list:")
print(nums1)
print(nums2)

result = map(lambda x, y: x + y, nums1, nums2)
print("Result: after adding two list")
print(list(result))
```

output

Program Name: Find numbers divisible by nineteen or thirteen from a list
demo81.py

```
nums = [19, 65, 57, 39, 152, 639, 121, 44, 90, 190]
print("Orginal list:")
print(nums)

result = list(filter(lambda x: (x % 19 == 0 or x % 13 == 0), nums))
print("Numbers of the above list divisible by nineteen or
thirteen:")
print(result)
```

output

Program Name: Extract numbers from string
demo82.py

```
str1 = "sdf 23 safs8 5 sdfsd8 sdfs 56 21sfs 20 5"
str_num = [i for i in str1.split(' ')]

num_str = sorted([x for x in str_num if x.isdigit()])
numbers = sorted([int(x) for x in str_num if x.isdigit()])

print(str1)
print(str_num)
print(num_str)
print(numbers)
```

output

Program Name: Sum of the positive and negative numbers of a given list
demo83.py

```
nums = [2, 4, -6, -9, 11, -12, 14, -5, 17]
print("Original list:", nums)

total_negative_nums = list(filter(lambda nums: nums<0,nums))
total_positive_nums = list(filter(lambda nums: nums>0,nums))

t_n = sum(total_negative_nums)
t_p = sum(total_positive_nums)

print("Sum of the positive numbers: ", t_n)
print("Sum of the negative numbers: ", t_p)
```

output

Program Name: Remove all elements from a given list present in another list
demo84.py

```
def m1(list1, list2):
        result = list(filter(lambda x: x not in list2, list1))
        return result

a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
b = [2, 4, 6, 8]

print(m1(a, b))
```

output

Program Name: Remove all elements from a given list present in another list
demo85.py

```
def m2(p):
        result = list(map(lambda x: "".join(reversed(x)), p))
        return result

a = ["Red", "Green", "Blue", "White", "Black"]
b = m2(a)
print(b)
```

output

Program Name: Count the occurrences of items in a given list
demo86.py

```
def count_occurrences(nums):
    result = dict(map(lambda el  : (el, list(nums).count(el)),
    nums))
    return result

nums = [3, 4, 5, 8, 0, 3, 8, 5, 0, 3, 1, 5, 2, 3, 4, 2]

print(nums)
print(count_occurrences(nums))
```

output

Program Name: Remove specific words from a given list using lambda.
demo87.py

```
def remove_words(list1, r_words):
    result = list(filter(lambda word: word not in r_words,
    list1))
    return result

colors = ['orange', 'red', 'green', 'blue', 'white', 'black']
remove_colors = ['orange', 'black']

print(colors)
print(remove_colors)
print(remove_words(colors, remove_colors))
```

output