

## UNIT III (GE8151 PROBLEM SOLVING AND PYTHON PROGRAMMING)

**3.6. Illustrative programs: square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.**

### **ILLUSTRATIVE EXAMPLES**

#### **1. Program to find the square root using Newton Method.**

```
def newtonSqrt(n, howmany):
```

```
    approx = 0.5 * n
```

```
    for i in range(howmany):
```

```
        betterapprox = 0.5 * (approx + n/approx)
```

```
        approx = betterapprox
```

```
    return betterapprox
```

```
print(newtonSqrt(10, 3))
```

```
print(newtonSqrt(10, 5))
```

```
print(newtonSqrt(10, 10))
```

#### **OUTPUT :**

```
Newton Sqrt Value is =.3.16231942215
```

```
Newton Sqrt Value is .=3.16227766017
```

```
Newton Sqrt Value is .=3.16227766017
```

#### **2. Program to find the GCD of two numbers**

```
d1=int(raw_input("Enter a number:"))
```

```
d2=int(raw_input("Enter another number"))
```

```
rem=d1%d2
while rem!=0 :
    d1=d2
    d2=rem
    rem=d1%d2
print "gcd of given numbers is : %d" %(d2)
```

**OUTPUT :**

```
Enter a number:54
Enter another number :24
GCD of given number is: 6
```

**3.Program to find the exponential of a number**

```
def power(base,exp):
    if(exp==1):
        return(base)
    if(exp!=1):
        return(base*power(base,exp-1))
base=int(input("Enter base: "))
exp=int(input("Enter exponential value: "))
print("Result:",power(base,exp))
```

**OUTPUT :**

```
Enter the base:3
Enter exponential value:2
Result: 9
```

#### **4.Program to find the sum of array of numbers**

```
arr = [1, 2, 3, 4, 5];
sum = 0;
for i in range(0, len(arr)):
    sum = sum + arr[i];
    print("Sum of all the elements of an array: " + str(sum));
```

#### **OUTPUT :**

Sum of all the elements of an array:15

#### **5.Program to find the maximum and minimum in a list**

```
list=[]
print("Enter the limit")
n=int(input())
print("Enter numbers")
for i in range(0,n):
    a=int(input())
    list.append(a)
maxno=list[0]
minno=list[0]
for i in range(len(list)):
    if list[i]>maxno:
        maxno=list[i]
    if list[i]<minno:
        minno=list[i]
```

```
print("Maximum no of the list",maxno)
```

```
print("Minimum no of the list",minno)
```

### **OUTPUT :**

Enter the limit:5

Enter numbers:1 2 3 4 5

Maximum no of the list:5

Minimum no of the list:1

### **6.Program to perform the linear search**

```
list = []
```

```
n=int(input("enter the no of elements in list"))
```

```
for i in range(0,n):
```

```
    a=int(input("enter the list elements"))
```

```
    list.append(a)
```

```
x = int(input("Enter number to search: "))
```

```
found = False
```

```
for i in range(0,n):
```

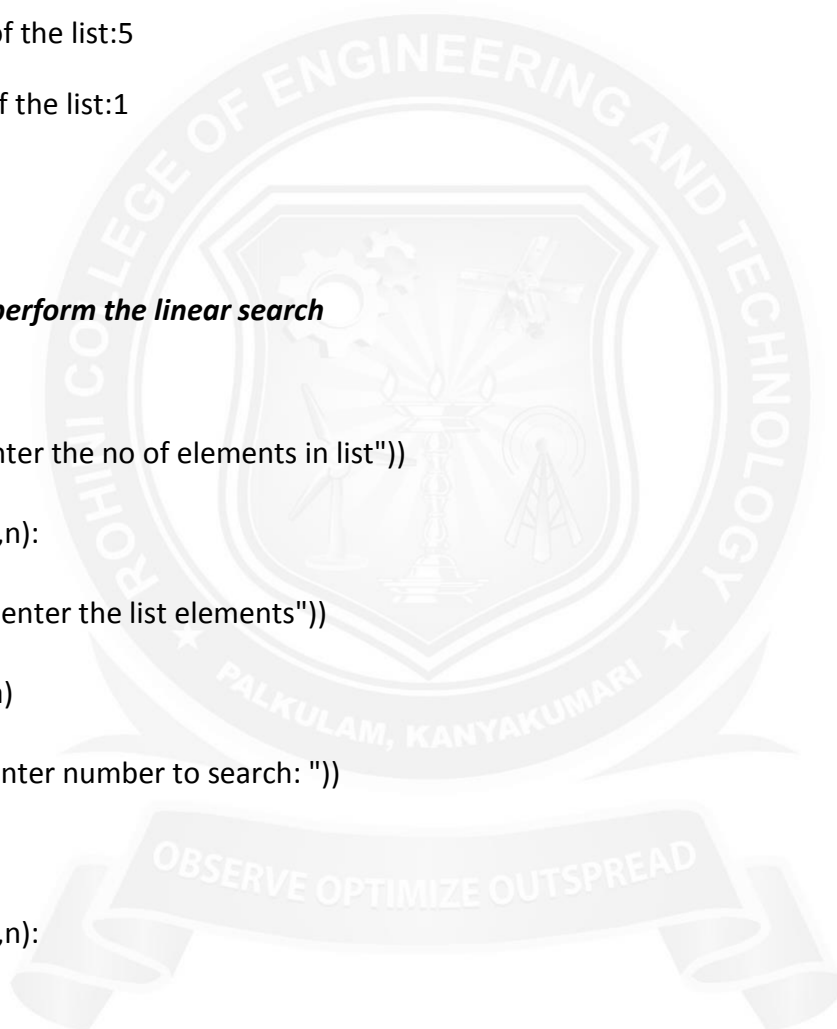
```
    if(list[i] == x):
```

```
        found = True
```

```
        print("%d found at %d position"%(x,i+1))
```

```
        break
```

```
if(found == False):
```



```
print("%d is not in list"%x)
```

### **OUTPUT :**

enter the no of elements in list:5

enter the list elements:1 2 3 4 5

Enter number to search:2

2 found at 1 position

### **7.Program to perform Binary search**

```
def binary_search(list,n,x):
```

```
    start = 0
```

```
    end = n - 1
```

```
    while(start <= end):
```

```
        mid = (start + end)/2
```

```
        if (x == list[mid]):
```

```
            return mid
```

```
        elif(x < list[mid]):
```

```
            end = mid - 1
```

```
        else:
```

```
            start = mid + 1
```

```
    return -1
```

```
n = input("Enter the size of the list: ")
```

```
list = []
```

```
for i in range(n):  
    list.append(input("Enter %d element: "%i))  
  
x = input("Enter the number to search: ")  
  
position = binary_search(list, n, x)  
  
if(position != -1):  
    print("Entered number %d is present at position: %d"%(x,position+1))  
  
else:  
    print("Entered number %d is not present in the list"%x)
```

**OUTPUT :**

```
Enter the size of the list:5  
Enter 1 element:1  
Enter 2 element:2  
Enter 3 element:3  
Enter 4 element:4  
Enter 5 element:5  
Enter the number to search:7  
Entered number 7 is not present in the list
```

## **ADDITIONAL PROGRAMS**

### **1. Write a python Program to Check if a Number is Positive, Negative or 0**

#### **Using if...elif...else**

```
num = float(input("Enter a number: "))
if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")
```

#### **Output:**

```
>>> Enter a number: 5
Positive number
>>>
```

#### **Using Nested if**

```
num = float(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
```

```
        print("Positive number")
    else:
        print("Negative number")
```

**Output:**

```
>>> Enter a number: 5
      Positive number
>>>
```

**2. Write a Python Program to Check a year is Leap Year or not.**

```
year = int(input("Enter a year: "))          # To get year (integer input) from the user
if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} is a leap year".format(year))
        else:
            print("{0} is not a leap year".format(year))
    else:
        print("{0} is a leap year".format(year))
else:
    print("{0} is not a leap year".format(year))
```

**Output:**

```
>>>
Enter a year: 2000
2000 is a leap year
>>>
Enter a year: 1991
1991 is not a leap year
```



### 3. Write a Python Program to Print the Fibonacci sequence

```
nterms = int(input("How many terms? "))  
  
# first two terms  
n1 = 0  
n2 = 1  
count = 0  
  
# check if the number of terms is valid  
if nterms <= 0:  
    print("Please enter a positive integer")  
elif nterms == 1:  
    print("Fibonacci sequence upto",nterms,":")  
    print(n1)  
else:  
    print("Fibonacci sequence upto",nterms,":")  
    while count < nterms: # Starting of While loop  
        print(n1,end=' , ')  
        nth = n1 + n2  
        # update values  
        n1 = n2  
        n2 = nth  
        count += 1 # Ending of While loop
```

#### **Output:**

How many terms? 10

Fibonacci sequence upto 10 :

0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 ,

```
>>>
```

#### 4. Write a Python Program to Check a number is Armstrong Number or not.

```
num = int(input("Enter a number: "))
sum = 0 # initialize sum
temp = num # find the sum and cube of each digit
while temp > 0:
    digit = temp % 10
    sum += digit ** 3
    temp //= 10
if num == sum: # display the result
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")
```

#### **Output:**

```
>>> Enter a number: 121
121 is not an Armstrong number
>>>
```

#### 5. Write a Python Program to Find LCM of two numbers

```
def lcm(x, y):
    if x > y:
        greater = x
    else:
        greater = y
    while(True):
        if((greater % x == 0) and (greater % y == 0)):
```

```

        lcm = greater
        break
    greater += 1

    return lcm

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print("The L.C.M. of", num1,"and", num2,"is", lcm(num1, num2))

```

**Output:**

```

>>> Enter first number: 10
      Enter second number: 15
      The L.C.M. of 10 and 15 is 30
>>>

```

**6. Write a Python Program to Add Two Matrices**

```

X = [[12,7,3],
     [4 ,5,6],
     [7 ,8,9]]
Y = [[5,8,1],
     [6,7,3],
     [4,5,9]]
result = [[0,0,0],
          [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[i][j] = X[i][j] + Y[i][j]

```

```

for r in result:

```

```
print(r)
```

**Output:**

```
>>>
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]
>>>
```

**7. Write a Python Program to Transpose a Matrix**

```
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]
>>>
```

## 8. Python Program to Multiply Two Matrices

```
# 3x3 matrix
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]]

# iterate through rows of X
for i in range(len(X)):
    # iterate through column Y
    for j in range(len(Y[0])):
        # iterate through rows of Y
        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]
```

## 9. Write a Python Program to Check Whether a String is Palindrome or Not

```
my_str = 'madame'
```

```

my_str = my_str.casefold()           # it suitable for caseless comparison
rev_str = reversed(my_str)          # reverse the string
if list(my_str) == list(rev_str):   # check the string is equal to its reverse
    print("It is palindrome")
else:
    print("It is not palindrome")

```

**Output:**

```

>>>
It is not palindrome
>>>

```

**10. Write a Python Program to count the number of each vowel in a string.**

```

vowels = 'aeiou'                     # string of vowels
ip_str = 'Hello, have you tried our tutorial section yet?'
                                     # change this value for a different result

ip_str = input("Enter a string: ")
ip_str = ip_str.casefold()           # make it suitable for caseless comparisons

count = {}.fromkeys(vowels,0)        # make a dictionary with each vowel a key and value 0

for char in ip_str:                  # count the vowels
    if char in count:
        count[char] += 1

print(count)

```

**Output:**

```

>>>{'o': 5, 'i': 3, 'a': 2, 'e': 5, 'u': 3}

```