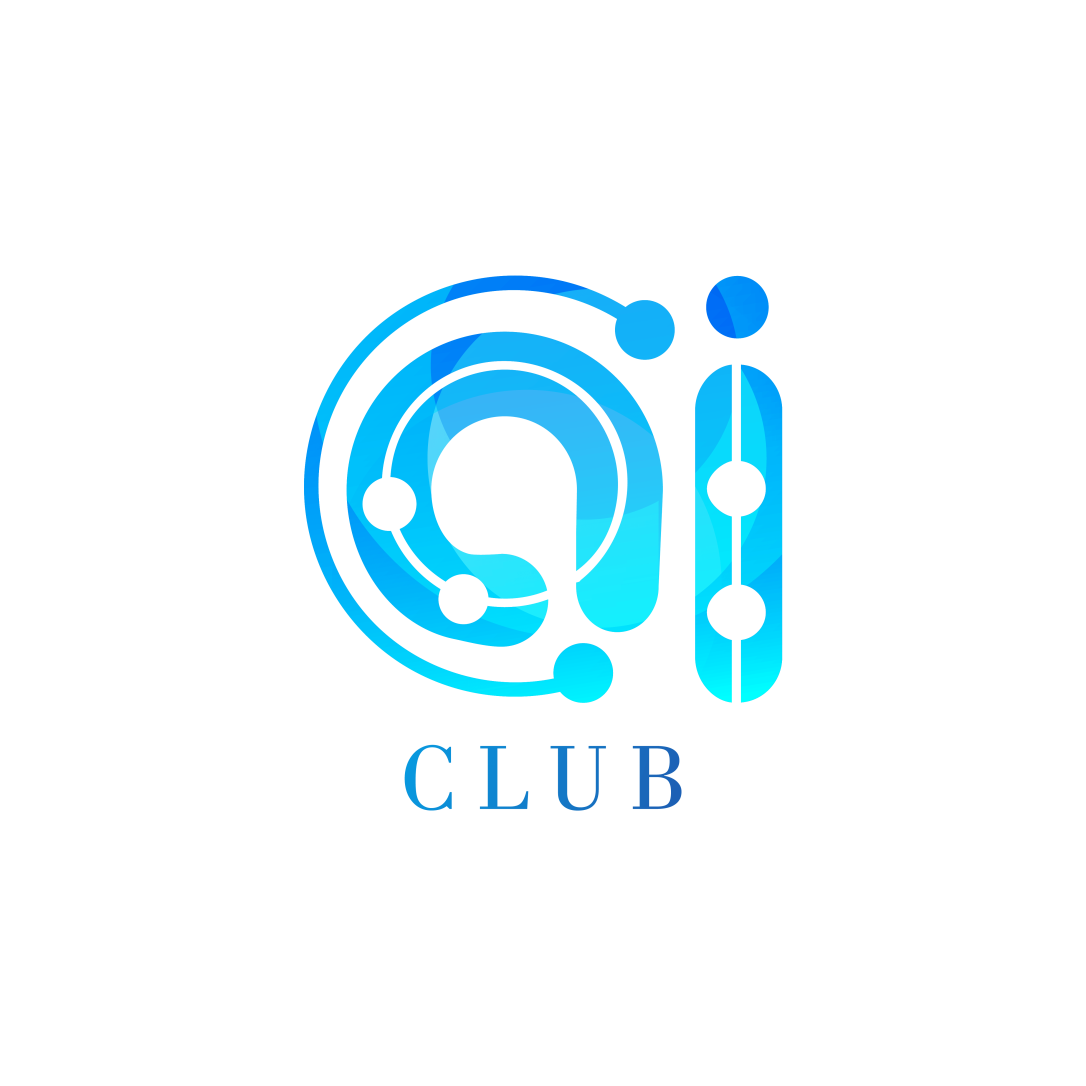
**Department of Artificial Intelligence and Machine Learning**



Laboratory Manual

Core Practical IV: Python Programming Lab

|  |  |
| --- | --- |
| Semester & Branch | III Semester – II AI&ML |
| Academic Year | 2024-2025 |
| Prepared By | Sheethal.P |

Approved by:

**Table of contents**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Title of contents** | **Page No.** |
|  | **Vision of the Institution** | **3** |
|  | **Mission of the Institution** | **3** |
|  | **Vision of the Department** | **4** |
|  | **Mission of the Department** | **4** |
|  | **Computer Lab Do’s and Don’ts** | **5** |
|  | **List of Programs / Experiments** | **6** |

**Department of Artificial Intelligence and Machine Learning**

**Vision of the Institution**

Towards building a dynamic learning community with ethical values, a kindled sense of environmental sustainability, and societal service, to meet global demands and challenges.

**Mission of the Institution**

The institution is committed to transform lives and serve society through a quest for excellence in teaching, innovation, lifelong learning, cultural enrichment, and outreach services.

To achieve its Vision and Mission, the College will endeavour:

* To create an intellectually inspiring, academically challenging culture of teaching and leading-edge research conducive to academic/professional excellence and lifelong learning for the learner and learning process.
* To provide holistic benchmarked education using state-of-art facilities in a participative learning environment for inculcating in-depth knowledge through accredited programs.
* To create a collaborative environment for a free exchange of ideas where creativity, innovation and Entrepreneurship flourish.
* To produce industry-ready graduates by imparting value-added programs, skill development courses, improved industry institution interactions, and enhanced placement activities.
* To inculcate, in our students, a deep sense of social responsibility, concern towards environmental sustainability & development, and communal harmony that transform them into socially responsible citizens.

**Vision of the Department**

* To Develop Artificial Intelligence and Machine Learning Specialists who can serve in almost all ranges of employment.

**Mission of the Department**

M1: To strive for excellence in development and deployment of Artificial Intelligence and Machine Learning.

M2: To develop young professionals with active and creative minds, a sense of understanding and compassion for others, and courage to act on their beliefs.

M3: To produce self-reliant, self-motivated and socially sensitive young AI graduates who can cater to the requirements of academia, industry, and research.

M4: To provide opportunities for Life Long Learning in the field of artificial intelligence.

**Laboratory Do’s and Don’ts**

**Do’s:-**

1. Remove your shoes or wear foot socks before you enter the lab
2. Always keep quiet. **Be considerate to other lab users.**
3. Report any problems with the computer to the person in charge.
4. Shut down the computer properly.
5. **Do not change computer settings or backgrounds.**

**Don’ts:-**

1. Do not bring any food or drinks in the computer room.
2. Don’t damage, remove, or disconnect any labels, parts, cables or equipment.
3. Do not install or download any software or modify or delete any system files on any lab computers.
4. If you leave the lab, do not leave your personal belongings unattended.
5. If you leave the lab, do not leave your personal belongings unattended.

**Index**

**List of Programs / List of Experiments**

|  |  |  |
| --- | --- | --- |
| **S. No./**  **Exp. No.** | **Program Name /Title of the Experiment** | **Page no.** |
|  | BIODATA | 07 |
|  | BIGGEST NUMBER | 10 |
|  | SUM OF ELEMENTS | 13 |
|  | MATRIX ADDITION | 16 |
|  | GREATEST COMMON FACTOR | 21 |
|  | FACTORIAL | 24 |
|  | FIBONACCI SERIES | 27 |
|  | PRIME NUMBERS | 30 |
|  | SORTING - LIST,TUPLE AND STRING | 33 |
|  | CALCULATOR | 38 |
|  | LINEAR SEARCH AND BINARY SEARCH | 42 |
|  | RANDOM NUMBERS | 46 |

**Exp.No: 1 BIODATA**

**AIM** :-

Write a python program that displays the following information: Your name, Full address Mobile number, College name, Course subjects

**ALGORITHM :-**

Step 1- Start the process

Step 2 – Create the variables for holding name, address, mobile number, college name and

Subject

Step 3 – Fetch all the values for name, address and mobile number using input function

Step 3 – Convert mobile number in to integer and check whether the user entered mobile

number is having 10 digits or not

Step 4 – If the mobile number is not valid display a message “enter a valid mobile number ”

on the screen and get a valid mobile number from the user

Step 5 – Extract the values for college name and course name from the user

Step 6 – Display the details stored in the variables name, address, mobile number, college

name and course name using print statements

Step 7 - Run the program and display the output

Step 8 – Stop the process

**CODING :-**

name=input("Enter your name:")

address=input("Enter your permanent address:")

mob=input("Enter your 10 digit mobile number:")

mob1=int(mob)

count=0

while mob1>0:

count=count+1

mob1=mob1//10

if count==10:

print("Mobile number is valid! Proceed with the other details")

else:

mob=input("Please enter a valid mobile number:")

college=input("Enter your college name:")

sub=input("Enter your branch of study:")

print("\*\*\*\*\*Student Details\*\*\*\*\*")

print("Name:",name)

print("Address:",end=" ")

add=address.split()

for i in add:

print(i)

print("Mobile Number:",mob)

print("College Name:",college)

print("Branch of Study:",sub)

**OUTPUT :**

Enter your name : Appu

Enter your permanent address : Lotus(H) Mundur(PO) Kaplippara Palakkad Kerala Pin-678592

Enter your 10 digit mobile number : 9947034421

Mobile number is valid! Proceed with the other details

Enter your college name : AJK College of Arts And Science

Enter your branch of study : B.Sc Digital And Cyber Forensic Science

\*\*\*\*\*Student Details\*\*\*\*\*

Name: Appu

Address: Lotus(H)

Mundur(PO)

Kaplippara

Palakkad

Kerala

Pin-678592

Mobile Number: 9947034421

College Name: AJK College of Arts And Science

Branch of Study : B.Sc Digital And Cyber Forensic Science

**Result :** The above program has been executed successfully. And output verified.

**Exp.No: 2 BIGGEST NUMBER**

**AIM :-**

Write a python program to find the largest three integers using if-else and conditional operator

**ALGORITHM :-**

Step 1 – Start the process

Step 2 – Create the variables x,y and z for holding user entered values

Step 3 - Create a variable named “a” to fetch the user choice

Step 4 – If the user entered choice is digit 1 find the biggest among three numbers using if-

else statement

Step 5 – Check x is greater than y and z if this condition is satisfied print the value

stored in the variable x as result

Step 6 – If the condition specified in the if- block is evaluated to false means check the value

stored in the variable y is greater than x and z or not, if this condition is evaluated to

true print the value of y as result

Step 7 – If both conditions in specified in the if and elif block is evaluated to false means

print the value of variable z as result

Step 8 – If the user entered choice is 2 check the biggest among 3 numbers using conditional

operator

Step 9 - Store the biggest number in the variable “big” after evaluating the condition and

display the output using print statement

Step 10 – If user entered choice is not valid quit the program

Step 11 – Stop the process

**CODING :**

x=int(input("Enter the first value :"))

y=int(input("Enter the second value :"))

z=int(input("Enter the third value :"))

big=0

a=int(input("Biggest number among 3 numbers can be checked through\n 1)Using if else\n 2)Using Ternary Operator\n 3)Quit\n Enter your choice :"))

if a==1:

if (x>y) and (x>z):

print("Biggest Value is=",x)

elif(y>x) and (y>z):

print("Biggest Value is=",y)

else:

print("Biggest Value is=",z)

elif a==2:

big=(x if (x>y and x>z) else (y if (y>x and y>z) else z))

print("Biggest Value is=",big)

else:

exit()

**OUTPUT :**

Enter the first value :120

Enter the second value :390

Enter the third value :788

Biggest number among 3 numbers can be checked through

1)Using if else

2)Using Ternary Operator

3)Quit

Enter your choice :1

Biggest Value is= 788

**Result :** The above program has been executed successfully. And output verified.

**Exp.No : 3 SUM OF ELEMENTS**

**AIM :**-

Write a python program that asks the user to enter a series of positive numbers (The user should enter a negative number to signal the end of the series) and the program should display the numbers in order and their sum

**ALGORITHM :-**

Step 1 - Start the process

Step 2 - Create an empty list x and a variable n

Step 3- Create a global variable named sum and initialize it to zero

Step 4 - Use a for loop to collect the series of values entered by the user

Step 5 – Check the user entered value is greater than zero or not

Step 6 – If the value is greater than zero insert it in to the list x using append function

Step 7 – If the value is less than zero use a break statement to come out of the loop

Step 8 – Sort the list using sort function

Step 9- Traverse directly through the list elements using a for loop

Step 10 – Add each item to the variable sum

Step 11 – Display the list elements in ascending order and display the sum of elements

Step 12 – Stop the process

**CODING :**

x=[]

sum=0

n=int(input("How many values do you want to insert?"))

for i in range(1,n+1):

y=int(input("Enter the value:"))

if y>0:

x.append(y)

else:

x.append(y)

break

x.sort()

for i in x:

print(i,end=" ")

sum=sum+i

print("\nThe sum is=",sum)

**OUTPUT :**

How many values do you want to insert?5

Enter the value:12

Enter the value:17

Enter the value:21

Enter the value:43

Enter the value:-10

-10 12 17 21 43

The sum is= 83

How many values do you want to insert?5

Enter the value:1

Enter the value:12

Enter the value:-10

-10 1 12

The sum is= 3

**Result :** The above program has been executed successfully. And output verified.

**Exp.No : 4 MATRIX ADDITION**

**AIM :-**

Write a python program to find the product of two matrices

**ALGORITHM :-**

Step 1 - Start the process

Step 2 – Create 2 empty list namely a and b

Step 3 - Create a variable n and fetch an integer value to create N\*N matrix

Step 4 – Use a for loop to fetch the values to create the matrix

Step 5 – Append each value to a local list called r

Step 6 – Append each row to the list a

Step 7 – Use another for loop to fetch matrix values from the user and append that values to

list b

Step 8 – Create an empty nested list called p store the sum of matrices

Step 9 – Use a nested for loop to find the sum of matrices

Step 10 – Print the values stored in matrix p

Step 11- Stop the process

**CODING :**

a=[]

n=int(input("Enter an integer for creating 3\*3(row\*column) matrix"))

print("\*\*\*\*\*Enter the elements\*\*\*\*\*")

for i in range(n):

r=[]

for j in range(n):

r.append(int(input()))

a.append(r)

print("\*\*\*\*\*The matrix is\*\*\*\*\*")

for i in range(n):

for j in range(n):

print(a[i][j],end=" ")

print()

b=[]

n=int(input("Enter an integer for creating 3\*3(row\*column) matrix:"))

print("\*\*\*\*\*Enter the elements\*\*\*\*\*")

for i in range(n):

r=[]

for j in range(n):

r.append(int(input()))

b.append(r)

print("\*\*\*\*\*The matrix is\*\*\*\*\*")

for i in range(n):

for j in range(n):

print(b[i][j],end=" ")

print()

p=[[0,0,0],[0,0,0],[0,0,0]]

for i in range(len(a)):

for j in range(len(b[0])):

for k in range(len(b)):

p[i][j]+=a[i][k]\*b[k][j]

print("\*\*\*\*\*The resultant matrix is\*\*\*\*\*")

for l in range(len(p)):

for j in range(len(p)):

print(p[i][j],end=" ")

print()

**OUTPUT :**

Enter an integer for creating N\*N(row\*column) matrix:3

\*\*\*\*\*Enter the elements\*\*\*\*\*

1

2

3

4

5

6

7

8

9

\*\*\*\*\*The matrix is\*\*\*\*\*

1 2 3

4 5 6

7 8 9

Enter an integer for creating N\*N(row\*column) matrix:3

\*\*\*\*\*Enter the elements\*\*\*\*\*

9

8

7

6

5

4

3

2

1

\*\*\*\*\*The matrix is\*\*\*\*\*

9 8 7

6 5 4

3 2 1

\*\*\*\*\*The resultant matrix is\*\*\*\*\*

30 24 18

84 69 54

138 114 90

**Result :** The above program has been executed successfully. And output verified.

**Exp.No :5 GREATEST COMMON FACTOR**

**AIM :-**

Write recursive functions for GCD of two integers

**ALGORITHM :-**

Step 1 – Start the process

Step 2 - Create a user defined function named gcd with 2 parameters a and b to calculate the

greatest common divisor of 2 numbers

Step 3 - Fetch 2 integer values from the user to calculate the greatest common divisor

Step 4 – Make function call by passing 2 arguments

Step 5 – Divide a by b using modulo operator and store the remainder value in variable x

Step 6 – If the remainder value is 0 return the value of b otherwise the function make a call

itself and calculate the gcd of the 2 numbers and return the result to the function call

Step 7 – Display the output

Step 8 – Stop the process

**CODING :**

def gcd(a,b):

x=a%b

if x==0:

return b

else:

return gcd(b,x)

c=int(input("Enter the first integer value:"))

d=int(input("Enter the second integer value:"))

y=gcd(c,d)

print("The Greatest Common Divisor of two numbers is:",y)

**OUTPUT :**

Enter the first integer value:52

Enter the second integer value:12

The Greatest Common Divisor of two numbers is: 4

**Result :** The above program has been executed successfully. And output verified.

**Exp.No :6 FACTORIAL**

**AIM :-**

Write recursive functions for the factorial of positive integer

**ALGORITHM :-**

Step 1- Start the process

Step 2- Create a user defined function named factorial with a single parameter

Step 3 – Create a variable n and fetch an integer value from the user to calculate the factorial

of the entered number

Step 4 - Check whether the user entered value is greater than zero or not

Step 5 – If the value is less than zero display a message “Factorial can’t be calculated for

negative numbers” on the screen

Step 6 – If the number is equal to zero display the output “Factorial of 0 is 1”

Step 7 – If the user entered value is greater than zero make a function call by passing the

argument to it

Step 8 – Inside the function check the user entered value is 1 or not

Step 9 – If the value is 1 return the value of x as output otherwise the function make a call to

Itself and return the factorial of the entered number to the function call

Step 10 – Display the output

Step 11 - Stop the process

**CODING :**

def factorial(x):

if x==1:

return x

else:

return x\*factorial(x-1)

n=int(input("Enter an integer value for calculating the factorial:"))

if n<0:

print("Factorial can not be calculated for negative numbers!!!")

elif n==0:

print("The factorial of Zero is 1")

else:

m=factorial(n)

print("The factorial of ",n," is=",m)

**OUTPUT :**

Enter an integer value for calculating the factorial:5

The factorial of 5 is= 120

**Result :** The above program has been executed successfully. And output verified.

**Exp.No :7 FIBONACCI SERIES**

**AIM :-**

Write recursive functions for Fibonacci Sequence up to given number n

**ALGORITHM :-**

Step 1 - Start the process

Step 2 - Create a user defined function named Fibonacci

Step 3 – Create a variable n and get the number of terms to generate Fibonacci series from the

user

Step 5 – If the number of terms entered by the user is zero or less than zero display a message

“You should enter a positive value to generate fibonacci series” on the screen

Step 6 – If the number of terms is equal to or less than 1 return the value of parameter x as

result

Step 7 – If the number of terms is greater than 1 generate the fibonacci series and display the

output

Step 8 –Stop the process

**CODING :**

def fibonacci(x):

if x<=1:

return x

else:

return(fibonacci(x-1)+fibonacci(x-2))

n=int(input("Enter the number of terms to generate fibonacci series:"))

if n<=0:

print("You should enter a positive value to generate fibonacci series!!!")

else:

print("\*\*\*The Fibonacci Series\*\*\*")

for i in range(n):

print(fibonacci(i))

**OUTPUT :**

Enter the number of terms to generate fibonacci series:10

\*\*\*The Fibonacci Series\*\*\*

0

1

1

2

3

5

8

13

21

34

**Result :** The above program has been executed successfully. And output verified.

**Exp.No :8 PRIME NUMBERS**

**AIM :-**

Write recursive functions to display prime number from 2 to n

**ALGORITHM :-**

Step 1 - Start the process

Step 2 – Define a function called prime to find the prime numbers between the digit 2 to

user entered value

Step 3 – Make a function call by passing 2 arguments to it

Step 4 – If both the parameters are same then return 0 to the function call

Step 5 - Otherwise divide n by a using modulo operator and if the remainder value is zero

return the value 1 to the function call

Step 6 – The remainder value while diving n by a is not equal to zero then make a function

call again in the return statement

Step 7 – Display the output

Step 8 – Stop the process

**CODING :**

def prime(a,n):

if n==a:

return 0

else:

if n%a==0:

return 1

else:

return prime(a+1,n)

num=int(input("Enter an integer value:"))

print("Prime numbers between 2 to ",num," are:")

for i in range(2,num+1):

if(prime(2,i)==0):

print(i,end=" ")

**OUTPUT :**

Enter an integer value:15

Prime numbers between 2 to 15 are:

2 3 5 7 11 13

**Result :** The above program has been executed successfully. And output verified.

**Exp.No :9 SORTING - LIST,TUPLE AND STRING**

**AIM :-**

Write a python program to sort a given sequence: String, List and Tuple

**ALGORITHM :-**

Step 1- Start the process

Step 2 – Create two empty list namely x and y

Step 3 – Create a variable a and fetch the users choice digit 1 for sorting list 2 for Sorting

tuple and 3 for Sorting a string

Step 4 – If the user’s choice is digit 1 then ask the user to enter the total number of elements

to insert in to the list

Step 5 - Use a for loop all fetch all the list elements and insert it in to the empty list x using

The built-in function append

Step 6 – Use a nested for loop for traversing through the list elements and comparing the

Values

Step 7 – After sorting the list elements in ascending order display the output

Step 8 – If the user’s choice is digit 2 repeat the step 4 to step 7 and convert the sorted list in

to a tuple using the built-in function tuple()

Step 9 – If the user’s choice is 3 fetch a string from the user and convert it in to a list using

the built-in function list()

Step 10 - Repeat the steps 6 and 7 and display the sorted string using a for loop

Step 11- Stop the process

**CODING :**

a=int(input("1)Soring a list\n2)Sorting s tuple\n3)Sorting a string\nEnter a value:"))

if a==1:

x=[]

n=eval(input("Please enter the total number of elements:"))

for i in range(n):

val=eval(input("Enter the value="))

x.append(val)

for i in range(n):

for j in range(i+1,n):

if x[i]>x[j]:

temp=x[i]

x[i]=x[j]

x[j]=temp

print("\*\*\*\*\*The list elements after sorting is\*\*\*\*\*")

for i in x:

print(i,end=" ")

elif a==2:

y=[]

n=eval(input("Please enter the total number of elements:"))

for i in range(n):

val=eval(input("Enter the value="))

y.append(val)

for i in range(n):

for j in range(i+1,n):

if y[i]>y[j]:

temp=y[i]

y[i]=y[j]

y[j]=temp

tup=tuple(y)

print("\*\*\*\*\*The tuple elements after sorting is\*\*\*\*\*")

for i in tup:

print(i,end=" ")

elif a==3:

s=input("Enter a string:")

t=list(s)

for i in range(len(t)):

for j in range(i+1,len(t)):

if t[i]>t[j]:

temp=t[i]

t[i]=t[j]

t[j]=temp

print("\*\*\*\*\*The albhets after sorting is\*\*\*\*\*")

for i in t:

print(i,end=" ")

else:

quit()

**OUTPUT :**

1)Soring a list

2)Sorting s tuple

3)Sorting a string

Enter a value:1

Please enter the total number of elements:5

Enter the value=12

Enter the value=9

Enter the value=3

Enter the value=134

Enter the value=10

\*\*\*\*\*The list elements after sorting is\*\*\*\*\*

3 9 10 12 134

1)Soring a list

2)Sorting s tuple

3)Sorting a string

Enter a value:2

Please enter the total number of elements:4

Enter the value=3

Enter the value=12

Enter the value=21

Enter the value=5

\*\*\*\*\*The tuple elements after sorting is\*\*\*\*\*

3 5 12 21

1)Soring a list

2)Sorting s tuple

3)Sorting a string

Enter a value:3

Enter a string:Python

\*\*\*\*\*The albhets after sorting is\*\*\*\*\*

P h n o t y

**Result :** The above program has been executed successfully. And output verified.

**Exp.No: 10 CALCULATOR**

**AIM :-**

Write a python program to make a simple calculator

**ALGORITHM :-**

Step 1- Start the process

Step 2 – Create the user defined functions add(), sub(), mul(), div(), modulo(), integerdiv()

for performing addition, subtraction, multiplication, division, modulus operation

and integer division operation respectively

Step 3 – Create 2 variables namely x and y and fetch 2 values from the user

Step 3 - Create a variable n to store the user’s choice 1) addition 2) Subtraction

3) Multiplication 4) Division 5) Integer division 6) Modulus and 7) Exit

Step 4 - Based on the users choice make a call to the corresponding function by passing 2

arguments to it

Step 5 - If the user entered an invalid choice come out of the program by using built-in

function exit()

Step 6 - Stop the process

**CODING :**

def add(a,b):

return a+b

def sub(a,b):

return a-b

def mul(a,b):

return a\*b

def div(a,b):

return a/b

def integerdiv(a,b):

return a//b

def modulo(a,b):

return a%b

x=eval(input("Enter a value for x:"))

y=eval(input("Enter a value for y:"))

n=int(input("\n1.Add\n2.Subtraction\n3.Multiplication\n4.Division\n5.Integer Division\n6.Modulus\n7.Exit\nEnter your choice:"))

if n==1:

z=add(x,y)

print("Addition result is=",z)

elif n==2:

z=sub(x,y)

print("Subtraction result is=",z)

elif n==3:

z=mul(x,y)

print("Multiplication result is=",z)

elif n==4:

z=div(x,y)

print("Division result is=",z)

elif n==5:

z=integerdiv(x,y)

print("Integer division result is=",z)

elif n==6:

z=modulo(x,y)

print("Modulus result is=",z)

else:

exit()

**OUTPUT :**

Enter a value for x:10

Enter a value for y:5

1.Add

2.Subtraction

3.Multiplication

4.Division

5.Integer Division

6.Modulus

7.Exit

Enter your choice:3

Multiplication result is= 50

**Result :** The above program has been executed successfully. And output verified.

**Exp.No :11 LINEAR SEARCH AND BINARY SEARCH**

**AIM :-**

Write a python program for Linear Search and Binary Search

**ALGORITHM :-**

Step 1 - Start the process

Step 2 – Create 2 user defined functions namely linearsearch and binarysearch with needed

parameters

Step 3 – Create a variable named m and store the user’s choice in to it .1) Linear serach

2) Binary search and 3) Exit

Step 4 – Create another variable named element to store the value to be searched in the list

Step 5 - If the user choice is 1 make a function call to the linearsearch function by passing

the 3 arguments list, length of the list and element to be searched

Step 6 – Start the search from the first element and check whether they are equal or not

Step 7 - If the element is found , return the index position of that element as output

Step 8 - If the element is not found, return element is not present as output

Step 9 – If the user’s choice is 2 make a function call to the binarysearch() by passing 2

arguments list and element to be searched

Step 10 – Compare the element to be searched with the middle element, if matches with the

middle element, we return the mid

Step 11 – Else if element to be searched is greater than the mid element, then element can

only lie in right half subarray after the mid element. So we recur for the right half

Step 12 – Else element to be searched is smaller recur for the left half

Step 13 – Display the output

Step 14 – Stop the process

**CODING :**

def linearsearch(l,n,e):

for i in range(0,n):

if(l[i]==element):

return i

return -1

def binarysearch(l,e):

low=0

high=len(l)-1

mid=0

while low<=high:

mid=(high+low)//2

if l[mid]<e:

low=mid+1

elif l[mid]>e:

high=mid-1

else:

return mid

return -1

x=[1,23,3,4,6,7,8,9,10,11,13,15,100]

length=len(x)

m=int(input("1.Linear Search\n2.Binary Search\n3.Exit\nChoose any method:"))

element=int(input("Enter a value to be searched:"))

if m==1:

y=linearsearch(x,length,element)

if(y==-1):

print("Element not found!!!")

else:

print("Element found at the index:",y)

elif m==2:

y=binarysearch(x,element)

if(y==-1):

print("Element not found!!!")

else:

print("Element found at the index:",y)

else:

exit()

**OUTPUT :**

1.Linear Search

2.Binary Search

3.Exit

Choose any method:1

Enter a value to be searched:10

Element found at the index(Linear Method): 8

1.Linear Search

2.Binary Search

3.Exit

Choose any method:2

Enter a value to be searched:10

Element found at the index(Binary Method): 8

1.Linear Search

2.Binary Search

3.Exit

Choose any method:1

Enter a value to be searched:1000

Element not found!!!

1.Linear Search

2.Binary Search

3.Exit

Choose any method:2

Enter a value to be searched:1000

Element not found!!!

**Result :** The above program has been executed successfully. And output verified.

**Exp.No: 12 RANDOM NUMBERS**

**AIM :-**

Write a python program that writes a series of random numbers to a file from 1 to n and display

**ALGORITHM**

Step 1 - Start the process

Step 2 – import the module random to program using the keyword random

Step 3 –Create a file pointer f and open the text file named “random.txt” in write mode(“w”)

Step 4 – Create a variable n to fetch the number of random elements to be inserted to the file

from the user

Step 5 – Use a for loop to fetch the sequence of random numbers generated by the randint()

function

Step 6 – Convert the random number generated by then randint() function into string using

str() function and write the string to the file “random.txt” using write() function

Step 7 – Open the file “random.txt” in read mode(“r”) to read the content of the file

Step 8 - Read the content of the file “random.txt” using read() function and display the

output

Step 9 – Stop the process

**CODING :**

import random

f=open("random.txt","w")

n=int(input("How many random numbers you want to enter in file:"))

for i in range(n):

j=random.randint(1,30)

j=str(j)

f.write(j)

f.write("\n")

f=open("random.txt","r")

str=f.read()

print("\*\*\*The random numbers are\*\*\*")

print(str)

**OUTPUT :**

How many random numbers you want to enter in file:10

\*\*\*The random numbers are\*\*\*

25

15

7

7

30

16

20

11

14

5

**Result :** The above program has been executed successfully. And output verified.