Ex. No:		EXCHANGE THE VALUES OF TWO VARIABLES
Date:		

AIM:

To develop a python program to check the swapping of two numbers

ALGORITHM:

Step 1: Start Process.
Step 2: Read a and b from User Input.
Step 3: Assign Variable temp as 0.
Step 4: Assign value of Variable a to the Variable temp.
Step 5: Assign value of Variable b to the Variable a.
Step 6: Assign value of Variable temp to the variable b.
Step 7: Display a and b.
Step 8: Stop Process.

PSEUDOCODE:

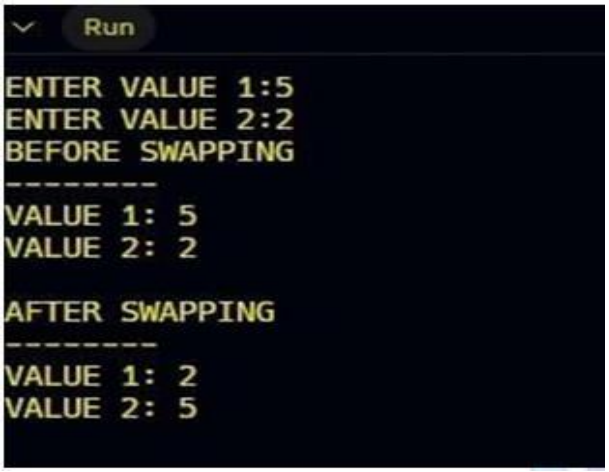
```
START
INPUT a,b
SET c=0
SET c=a, a=b, b=c
PRINT a, b
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
a=int (input ("Enter Value-1:"))
b=int (input ("Enter Value 2:"))
print ("BEFORE SWAPPING","\n -------")
print ("Value-1:”, a)
print ("Value-2:”, b)
c=0
c=a
a=b
b=c
print ("AFTER SWAPPING","\n--------")
print ("Value-1:”, a)
print ("Value-2:”, b)
```

OUTPUT:



RESULT:

Thus, the python program to swap the values of two number has been developed, executed and tested successfully.

Ex. No:		CIRCULATE THE VALUES OF N VARIABLES
Date:		

AIM:

To develop a python program to check the circulate the values of n variables.

ALGORITHM:

Step 1: Start Process.
Step 2: Assign the value a to 1.
Step 3: Assign the value b to 2.
Step 4: Assign the value c to 3.
Step 5: Assign the value d to 4.
Step 6: Assign the value a to temp
Step 7: Assign the value d to a.
Step 8: Assign the value c to d.
Step 9: Assign the value b to c.
Step 10: Assign the value temp to b.
Step 11: Display a, b, c and d.
Step 12: Stop Process.

PSEUDOCODE:

```
START
INIT a=1 b=2 c=3 d=4 temp=0
SET temp=a a=d d=c c=b b=temp
PRINT a, b, c, d
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
a = 1
b = 2
c = 3
d = 4
print ("Before circulate")
print ("----------------")
print ("a= “, a,"b= “, b,"c= “, c,"d= “, d)
temp=a
a=d
d=c
c=b
b=temp
print ("After circulate")
print ("----------------")
print ("a= “, a,"b= “, b,"c= “, c,"d= “, d)
```

OUTPUT:

RESULT:

Thus, the python program to circulate n variables has been developed, executed and tested successfully.

Ex. No:		FIND THE DISTANCE BETWEEN TWO POINTS
Date:		

AIM:

To develop a python program to find the distance between two points.

ALGORITHM:

Step 1: Start Process.
Step 2: Read x1, x2, y1, y2 from User Input.
Step 3: Compute (x2-x1) **2 and store it in term1.
Step 4: Compute (y2-y1) **2 and store it in term2.
Step 5: Compute (term1+term2) **0.5 and store it in ans
Step 6: Display "The Distance is ", ans
Step 7: Stop Process.

PSEUDOCODE:

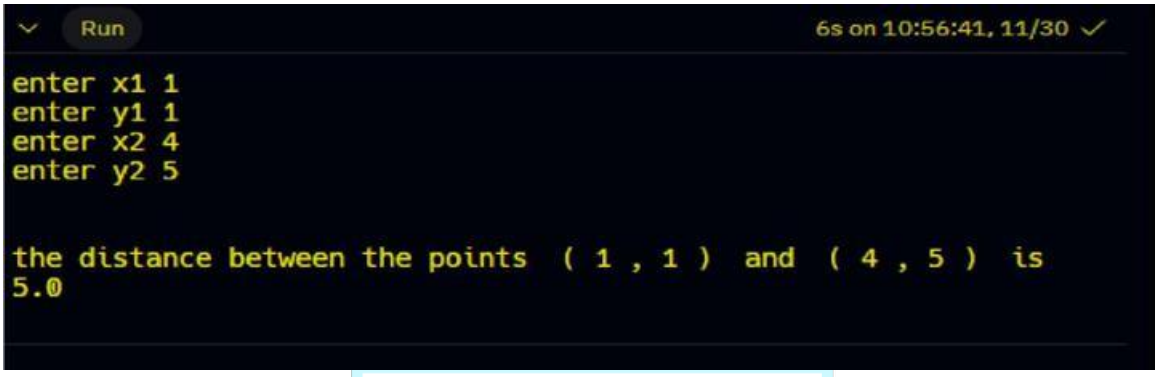
```
START
READ x1, x2, y1, y2
ASSIGN term1=(x2-x1) **2
ASSIGN term2=(y2-y1) **2
ASSIGN ans =(term1+term2) **0.5
DISPLAY "The Distance is ", ans
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
x1=int (input ("enter x1 "))
y1=int (input ("enter y1 "))
x2=int (input ("enter x2 "))
y2=int (input ("enter y2 "))
term1=(x2-x1) **2 term2=(y2-y1) **2
ans=(term1+term2) **0.5
print (("The Distance between the points","(“, x1,",”
y1,")","and","(“, x2,",”, y2,")"," is “, round(ans,2))
```

OUTPUT:



RESULT:

Thus, the python program to find distance between two points has been developed, executed and tested successfully.

Ex. No:		PRINT A NUMBER PATTERN 1
Date:		

AIM:

To develop a python program to Print a Number Pattern 1

ALGORITHM:

Step 1: Start Process.
Step 2: Read n.
Step 3: For i in the range from 1 to n+1
    Step 3.1: For j in the range from 1 to i+1
        Step 3.1.1: Display j
    Step 3.2: Display new line
Step 4: Stop Process.

PSEUDOCODE:

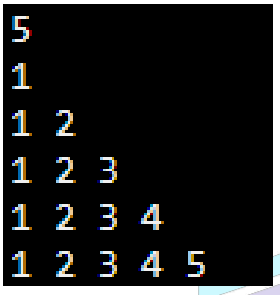
```
START
INPUT n
FOR i in range 1, n+1
     FOR j in range 1, i+1
          PRINT j, end = “ ”
     PRINT
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
n=int (input ())
for i in range (1, n+1):
    for j in range (1, i+1):
        print (j, end=" ")
    print ()
```

OUTPUT:



RESULT:

Thus, the python program to Print a number Pattern 1 has been developed, executed and tested successfully.

Ex. No:		PRINT A NUMBER PATTERN 2
Date:		

AIM:

To develop a python program to Print a Number Pattern 2

ALGORITHM:

Step 1: Start Process.
Step 2: Read n.
Step 3: For i in the range from 1 to n+1
  Step 3.1: For k in the range from 1 to i-1
    Step 3.1.1: Display one empty space
  Step 3.2: For j in the range from 1, n+1
    Step 3.2.1: Display j
  Step 3.3: Display new line
  Step3.4: Decrement n by 1
Step 4: Stop Process.

PSEUDOCODE:

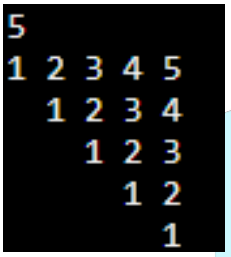
```
START
FOR i in range 1 to n+1
     FOR k in range 1 to i-1
          PRINT " ", end = " "
     FOR j in range 1 to i+1
          PRINT j, end = " "
     PRINT
     n-=1
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
n=int (input ())
for i in range (1, n+1):
    for k in range (0, i-1):
       print (‘ ’, end=" ")
    for j in range (1, n+1):
        print (j, end=" ")
    print ()
    n-=1
```

OUTPUT:



RESULT:

Thus, the python program to Print the Number Pattern 2 has been developed, executed and tested successfully.

Ex. No:		PRINT A PYRAMID PATTERN 1
Date:		

AIM:

To develop a python program to Print a Pyramid Pattern 1

ALGORITHM:

Step 1: Start Process.
Step 2: Read n.
Step 3: For i in the range from n to 0 step value as -1
    Step 3.1: For j in the range from I to 0 step value as -1
        Step 3.1.1: Display "*"
    Step 3.2: Display New Line
Step 4: Stop Process.

PSEUDOCODE:

```
START
FOR i in range n, 0, -1
     FOR j in range i, 0, -1
          PRINT '*', end = " "
     PRINT
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
n=int (input ())
for i in range (n,0, -1):
    for j in range (i,0, -1):
        print ('*', end=" ")
    print ()
```

OUTPUT:

RESULT:

Thus, the python program to Print a Pyramid Pattern 1 has been developed, executed and tested successfully.

.

Ex. No:		PRINT A PYRAMID PATTERN 2
Date:		

AIM:

To develop a python program to Print a Pyramid Pattern 2.

ALGORITHM:

Step 1: Start Process.
Step 2: Read a and b.
Step 3: For i in the range from 0 to n+1 step value as 2
  Step 3.1: For j in the range from 0 to n-i-1
    Step 3.1.1: Display " "
  Step 3.2: For k in the range from 0 to i+1
    Step 3.2.1: Display "*"
  Step 3.3: Display New line
Step 4: Stop Process.

PSEUDOCODE:

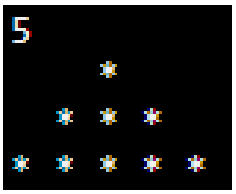
```
START
FOR I in range 0, n+1, 2
     FOR j in range 0, n-i-1
          PRINT end= " "
     FOR k in range 0, i+1
          PRINT "*"
     PRINT
STOP
```

FLOW CHART:

IMPLEMENTATION CODE:

```
n=int (input ())
for i in range (0, n+1,2):
     for j in range (0, n-i-1):
          print (end=" ")
     for k in range (0, i+1):
          print ("*", end= " ")
     print ()
```

OUTPUT:



RESULT:

Thus, the python program to Print a Pyramid Pattern 2 has been developed, executed and tested successfully.

.

Ex. No:		FACTORIAL OF A NUMBER
Date:		

AIM:

To develop a python function that accepts a number & returns factorial value of the number.

ALGORITHM:

Step 1: Start Process.
Step 2: Read a and b.
Step 3: Assign temp=0.
Step 4: Assign value of a to the variable temp.
Step 5: Assign value of b to the variable a.
Step 6: Assign value of temp to the variable b.
Step 7: Display a and b.
Step 8: Stop Process.

IMPLEMENTATION CODE:

```
def factorial(x):
    fact=1
    for i in range (1, x+1):
        fact*=i
    return fact
n=int (input ("Enter Number: "))
print ("Factorial is", factorial(n))
```

OUTPUT:

CASE – 1:

```
Enter Number: 1
Factorial is 1
```

CASE - 2

```
Enter Number: 3
Factorial is 6
```

CASE – 3:

```
Enter Number: 5
Factorial is 120
```

RESULT:

Thus, the python program to a python function that accepts a number & returns factorial value of the number.

Ex. No:		LARGEST IN A GIVEN LIST
Date:		

AIM:

To develop a python function that accepts a list & returns largest in the list

ALGORITHM:

#__main ()

Step 1: Start Process.
Step 2: Read l from user input.
Step 3: Display "The maximum of given list is:", largest_num(l)
Step 4: Stop Process.

#largest_num(l)

Step 1: Start Function.
Step 2: Assign max variable with 0.
Step 3: if l == [], return -1 else go to next step.
Step 4: For i in l
Step 4.1: if I > max, assign max as i
Step 5: return max
Step 4: Stop Function.

IMPLEMENTATION CODE:

```
def largest_num(l):
    max=0
    if l== []:
        return -1
    else:
        for i in l:
            if i>max:
                max=i
    return max
```

```
l=eval (input ("Enter list Of Numbers: "))
print ("Maximum of Given List is:”, largest_num(l))
```

OUTPUT:

CASE – 1:

```
Enter list Of Numbers: 10,11,1,4,5,13,2
Maximum of Given List is: 13
```

CASE – 2:

```
Enter list Of Numbers: 2,3,1,4,5
Maximum of Given List is: 5
```

RESULT:

Thus, python function that accepts a list & returns largest in the list has been developed, executed and tested successfully.

Ex. No:		AREA OF THE SHAPES
Date:		

AIM:

To develop a python function for finding the area for following shapes:

1.Square
2.Rectangle
3.Triangle
4.Circle

ALGORITHM:

#__main__()

Step 1: Start Process
Step 2: Read c from user input.
Step 3: if c is equal to 1 display area by calling function area_sqr()
Step 4: if c is equal to 2 display area by calling function area_rect()
Step 5: if c is equal to 3 display area by calling function area_tri()
Step 6: if c is equal to 4 display area by calling function area_circle()
Step 7: else for other choices display "invalid choice"
Step 8: Stop Process

#area_sqr(s)

Step 1: Start Function.
Step 2: return s*s
Step 3: Stop Function

#area_rect(l, b)

Step 1: Start Function.
Step 2: return l*b.
Step 3: Stop Function.

#area_tri(b, h)

Step 1: Start Function
Step 2: return 0.5*b*h
Step 3: Stop Process.

#area_circle(r)

Step 1: Start Function.
Step 2: Return 3.14*r*r
Step 3: Stop Function.

IMPLEMENTATION CODE:

```
def area_sqr(a):
    return a*a
def area_rect (l, b):
    return l*b
def area_tri (b, h):
    area=0.5*b*h
    return area
def area_circle(r):
    area=3.14*(r)**2
    return round(area)
print ("AREA OF SHAPES")
print ("1. Square")
print ("2. Rectangle")
print ("3. Triangle")
print ("4. Circle")
while True:
    c=int (input ("Enter Your choice: "))
    if c==1:
        s=int (input ("Enter Side of Square: "))
        print ("The Area of Square is:", area_sqr(s))
    elif c==2:
        l=int (input ("Enter Length of Rectangle: "))
        b=int (input ("Enter Breadth of the
Rectangle: "))
        print ("The Area of Rectangle is:", area_rect
(l, b))
    elif c==3:
```

```
            b=int (input ("Enter Base of Triangle: "))
            h=int (input ("Enter Height of the Triangle:
    "))
            print ("The Area of Triangle is:", area_tri
    (b, h))

        elif c==4:
            r=int (input ("Enter Radius of Circle: "))
            print ("The Area of Circle is:",
    area_circle(r))
        else:
            print ("Enter Correct input")
        ans=input ("Do you Want to Continue? ('y' for
    continue or press other key for break): ")
        if ans.lower ()! ='y':
            break
```

OUTPUT:

CASE – 1:

```
AREA OF SHAPES
1.Square
2.Reactangle
3.Triangle
4.Circle
Enter Your choice: 1
Enter Side of Square: 5
The Area of Square is: 25
Do you Want to Continue?('y' for continue or press other key for break):
```

CASE – 2:

```
AREA OF SHAPES
1.Square
2.Reactangle
3.Triangle
4.Circle
Enter Your choice: 2
Enter Length of Rectangle: 5
Enter Breadth of the Rectangle: 4
The Area of Rectangle is: 20
Do you Want to Continue?('y' for continue or press other key for break):
```

RESULT:

Thus, python program python function for finding the area of given shapes has been developed, executed and tested successfully.

Ex. No:		REVERSE A GIVEN STRING
Date:		

AIM:

To develop a python program reverse a given string.

ALGORITHM:

Step 1: Start Process.
Step 2: Read str from user input.
Step 3: Assign rev_str as empty string.
Step 4: For i in the range from length of str-1 to -1 step value as -1
Step 4.1: Update rev_str=rev_str[i]
Step 5: Display "The Reversed string is:", rev_str
Step 6: Stop Process.

IMPLEMENTATION CODE:

```
str_1=input("Enter a string:")

print("The reversed of",str_1,"is",str_1[::-1])
```

OUTPUT:

CASE – 1:

```
Enter a String: bus
The Reversed String is: sub
```

CASE - 2:

```
Enter a String: driver
The Reversed String is: revird
```

CASE – 3:

```
Enter a String: passanger
The Reversed String is: regnassap
```

RESULT:

Thus, python program to reverse a given string has been developed, executed and tested successfully.

Ex. No:		PALLINDROME CHECK
Date:		

AIM:

To develop a python program to check the string is palindrome or not.

ALGORITHM:

Step 1: Start Process.
Step 2: Read str from user input.
Step 3: Assign rev_str to an empty string.
Step 4: For i in the range from length of str-1 to -1, step value as -1
Step 4.1: Update rev_str as rev_str+str[i]
Step 5: If rev_str = str go to next step else go to step 7
Step 6: Display "The string is palindrome."
Step 7: Display "The string is not pallindrome."
Step 8: Stop Process.

IMPLEMENTATION CODE:

```
str_1=input("Enter a string:")


str_2=str_1[::-1]

if str_1==str_2:

    print(str_1," is a palindrome")

else:

    print(str_1,"is not a palindrome")
```

OUTPUT:

CASE – 1:

```
Enter a String: bus
The Reversed String of bus is sub bus is NOT PALLINDROME
```

CASE -2:

```
Enter a String: driver
The Reversed String of driver is revird driver is NOT PALLINDROME
```

CASE – 3:

```
Enter a String: malayalam
The Reversed String of malayalam is malayalam malayalam is PALLINDROME
```

RESULT:

Thus, python program to check the string is palindrome or not has been developed, executed and tested successfully.

Ex. No:		CHARACTER COUNT
Date:		

AIM:

To develop a python program to count the number of specified characters in a given string

ALGORITHM:

Step 1: Start Process.
Step 2: Read str from user input.
Step 3: Read char from user input.
Step 4: Assign count variable as value 0
Step 5: For i in str
Step 5.1: If char is equals to I increment count by 1
Step 6: Display the string count.
Step 7: Stop Process.

IMPLEMENTATION CODE:

```
str_1=input("Enter a string:")

a=input("Enter the char to be counted:")

print("The string",str_1,"contains",str_1.count(a),a)
```

OUTPUT:

CASE – 1:

```
Enter a String: bus
Enter a Character to be counted: b
The String bus Contains 1 char
```

CASE – 2:

```
Enter a String: driver
Enter a Character to be counted: r
The String driver Contains 2 char
```

CASE – 3:

```
Enter a String: passangers
Enter a Character to be counted: s
The String passangers Contains 3 char
```

RESULT:

Thus, python program to count the number of specified characters in a given string has been developed, executed and tested successfully.

Ex. No:		REPLACE A SUBSTRING IN A STRING
Date:		

AIM:

To develop a python program to replace a sub string in the given string

ALGORITHM:

Step 1: Start Process.
Step 2: Read str from user input.
Step 3: Assign upd_str variable with empty string.
Step 4: Assign new_str variable with empty string.
Step 5: Add a count() method
Step 6: Display the updated string.
Step 7: Stop Process.

IMPLEMENTATION CODE:

```
str_1=input("Enter a string:")
a=input("Enter a char want to replace:")
b=input("Enter a char to be replaced:")
print("After the replacement the
string",str_1,"becomes",str_1.replace(a,b))
```

OUTPUT:

CASE -1:

```
Enter a String: bus
Enter the Character want to replace: b
Enter the Character want to be replaced: i
After Replacement the string bus becomes ius
```

CASE – 2:

```
Enter a String: driver
Enter the Character want to replace: r
Enter the Character want to be replaced: x
After Replacement the string driver becomes dxivex
```

CASE – 3:

```
Enter a String: passanger
Enter the Character want to replace: s
Enter the Character want to be replaced: y
After Replacement the string passanger becomes payyanger
```

RESULT:

Thus, python program to replace a sub string in the given string has been developed, executed and tested successfully.

Ex. No:		LIBRARY MANAGEMENT SYSTEM
Date:		

AIM:

To develop a Python program for a simple library management system.

ALGORITHM:

Step 1: Start Process.
Step 2: Assign library_list as empty list.
Step 3: Read choice from the user input from the given operations.
Step 4: Write the functions for adding, viewing, searching, updating, deleting the books in the library.
Step 5: Call the function for the respective choices by user.
Step 6: Ask the user weather they have to continue or not.
Step 7: Stop Process.

IMPLEMENTATION CODE:

```
"""

To Write a Program for Library Management System

"""

#functions()

def check_lib_empty(book_details):
    if book_details!=[]:
        return True
    else:
        return False


def view_book(book_details):
    if check_lib_empty(book_details)==True:
        for book in book_details:
            print('\n',book)
    else:
```

```
:
        print("\nLIBRARY IS EMPTY")


def add_book(book_details):
  while True:
      book_id = int(input("\nEnter ISBN ID: "))
      book_name = input("Enter Book Title: ")
      book_author = input("Enter Author's Name: ")
      book_qty = int(input("Enter No. of Quantity:
"))
      tup = (book_id, book_name, book_author,
book_qty)
      book_details.append(tup)
      tup = ()
      c = input("\nDo you Want to add more Books to
Library(Press Y to add Book or Press other key to
Quit): ")
      if c.lower() != 'y':
          break
  view_book(book_details)


def search_book(book_details):
    if check_lib_empty(book_details)==True:
        print("""\nEnter the details you know about
the book
    1.ISBN ID
    2.Book Name
    3.Book Author""")
        c=int(input("\nEnter Your choice: "))
        if c==1:
            book_id=int(input("Enter the ISBN ID: "))
            for book in book_details:
                if book[0]==book_id:
                    print(book)
                else:
                    print("\nBOOK NOT FOUND!")
        elif c==2:
            book_name=input("Enter Book Name: ")
            for book in book_details:
```

```
                if
book[1].lower()==book_name.lower():
                    print(book)
                else:
                    print("\nBOOK NOT FOUND!")
        elif c==3:
            book_author=input("Enter Author's Name:
")
            for book in book_details:
                if
book[2].lower()==book_author.lower():
                    print(book)
                else:
                    print("\nBOOK NOT FOUND!")
        else:
            print("\nPlease Give Correct Input")
    else:
        print("\nLIBRARY IS EMPTY")


def upd_det(book_details,book):
     book_details.remove(book)
     book_id=int(input("Enter Updated ISBN ID: "))
     book_name = input("Enter Updated Book Title: ")
     book_author = input("Enter Updated Author's
Name: ")
     book_qty = int(input("Enter Updated No. of
Quantity: "))
     tup = (book_id, book_name, book_author,
book_qty)
     book_details.append(tup)


def update_book(book_details):
    if check_lib_empty(book_details)==True:
        print("""\nEnter the details you know about
the book
    1.ISBN ID
    2.Book Name
    3.Book Author""")
        c=int(input("Enter Your choice: "))
        if c==1:
```

```
                book_id=int(input("Enter the ISBN ID: "))
                if book_details==[]:
                    print("\nLIBRARY IS EMPTY")
                else:
                    for book in book_details:
                        if book[0]==book_id:
                            upd_det(book_details,book)
                        else:
                            print("\nBOOK NOT FOUND!")
            elif c==2:
                book_name=input("Enter Book Name: ")
                if book_details==[]:
                    print("\nLIBRARY IS EMPTY")
                else:
                    for book in book_details:
                        if
book[1].lower()==book_name.lower():
                            upd_det(book_details,book)
                        else:
                            print("\nBOOK NOT FOUND!")
            elif c==3:
                book_author=input("Enter Author's Name:
")
                if book_details==[]:
                    print("\nLIBRARY IS EMPTY")
                else:
                    for book in book_details:
                        if
book[2].lower()==book_author.lower():
                            upd_det(book_details,book)
                        else:
                            print("\nBOOK NOT FOUND!")
            else:
                print("\nPlease Give Correct Input")
        else:
            print("\nLIBRARY IS EMPTY")
        view_book(book_details)


    def delete_book(book_details):
        if check_lib_empty(book_details)==True:
```

```
        print("""Enter the details you know about the
book
    1.ISBN ID
    2.Book Name
    3.Book Author""")
        c=int(input("\nEnter Your choice: "))
        if c==1:
            book_id=int(input("\nEnter the ISBN ID:
"))
            for book in book_details:
                if book[0]==book_id:
                    book_details.remove(book)
                else:
                    print("\nBOOK NOT FOUND!")
        elif c==2:
            book_name=input("Enter Book Name: ")
            for book in book_details:
                if
book[1].lower()==book_name.lower():
                    book_details.remove(book)
                else:
                    print("\nBOOK NOT FOUND!")
        elif c==3:
            book_author=input("Enter Author's Name:
")
            for book in book_details:
                if
book[2].lower()==book_author.lower():
                    book_details.remove(book)
                else:
                    print("\nBOOK NOT FOUND!")
        else:
            print("\nPlease Give Correct Input")
    else:
        print("\nLIBRARY IS EMPTY")
    view_book(book_details)

#__main__

print('-'*40)
print("LIBRARY MANAGEMENT SYSTEM")
print('-'*40)
```

```
library_list=list()

while True:
    print("""\nEnter the operations you want to do\n
1.ADD BOOK TO LIBRARY
2. IEW BOOKS IN LIBRARY
3.SEARCH BOOK IN LIBRARY
4.UPDATE BOOK IN LIBRARY
5.DELETE BOOK IN LIBRARY""")
    c=int(input("\nEnter Your Choice: "))
    if c==1:
        add_book(library_list)
    elif c==2:
        view_book(library_list)
    elif c==3:
        search_book(library_list)
    elif c==4:
        update_book(library_list)
    elif c==5:
        delete_book(library_list)
    choice=input("\nDO YOU WANT TO DO ANY OTHER
OPERATIONS(PRESS 'Y' TO CONTINUE OR PRESS ANY OTHER
KEY TO QUIT: ")
    if choice.lower()!='y':
        break
```

OUTPUT:

```
Enter the operations you want to do

1.ADD BOOK TO LIBRARY
2.VIEW BOOKS IN LIBRARY
3.SEARCH BOOK IN LIBRARY
4.UPDATE BOOK IN LIBRARY
5.DELETE BOOK IN LIBRARY

Enter Your Choice: 1

Enter ISBN ID: 123
Enter Book Title: Python
Enter Author's Name: Ramesh Theja
Enter No. of Quantity: 60

Do you Want to add more Books to Library(Press Y to add Book or Press
other key to Quit): g

 (123, 'Python', 'Ramesh Theja', 60)

DO YOU WANT TO DO ANY OTHER OPERATIONS(PRESS 'Y' TO CONTINUE OR PRESS
ANY OTHER KEY TO QUIT: y

Enter the operations you want to do

1.ADD BOOK TO LIBRARY
2.VIEW BOOKS IN LIBRARY
3.SEARCH BOOK IN LIBRARY
4.UPDATE BOOK IN LIBRARY
5.DELETE BOOK IN LIBRARY

Enter Your Choice: 2

 (123, 'Python', 'Ramesh Theja', 60)

DO YOU WANT TO DO ANY OTHER OPERATIONS(PRESS 'Y' TO CONTINUE OR PRESS
ANY OTHER KEY TO QUIT:
```

RESULT:

Thus, python program for a simple library management system has been developed, executed and tested successfully.

Ex. No:		COPY FROM ONE FILE TO ANOTHER
Date:		

AIM:

To develop a python program to copy the contents of the file from source.txt to destination.txt.

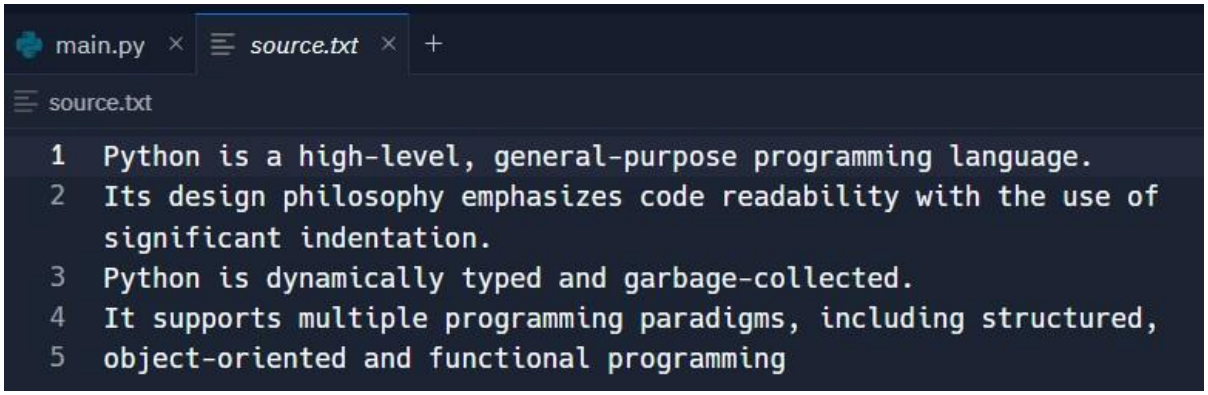
ALGORITHM:

Step 1: Start Process.
Step 2: Open the file source.txt as src_file
Step 3: Open the file destination.txt as dest_file
Step 4: Read the data in src_file in src_file_data
Step 5: Write the data in src_file_data in dest_file
Step 6: Close src_file
Step 7: Close dest_file
Step 8: Stop Process.

IMPLEMENTATION CODE:

```
src_file = open("source.txt", "r")
dest_file = open("destination.txt", "w")
src_file_data = src_file.read()
dest_file.write(src_file_data)
src_file.close()
dest_file.close()
```

OUTPUT:



main.py × destination.txt × +

destination.txt

```
1  Python is a high-level, general-purpose programming language.
2  Its design philosophy emphasizes code readability with the use of
   significant indentation.
3  Python is dynamically typed and garbage-collected.
4  It supports multiple programming paradigms, including structured,
5  object-oriented and functional programming
```

RESULT:

Thus, python program to copy the contents of the file from source.txt to destination.txt. has been developed, executed and tested successfully.

Ex. No:		WORD COUNT IN A FILE
Date:		

AIM:

To develop a python program to Count the number of words in a file.

ALGORITHM:

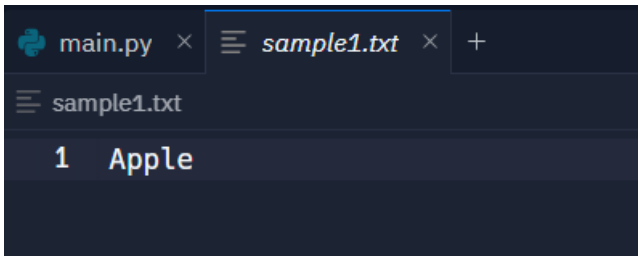
Step 1: Start Process.
Step 2: Read file name from user input.
Step 3: Open the file from user input.
Step 4: Read the data from the file as file_data
Step 5: Split the file_data into words.
Step 6: Assign file_count variable as 0.
Step 7: Use for loop to count the words in the variable words in file_count.
Step 8: Display the number of words in file.
Step 9: Stop Process.

IMPLEMENTATION CODE:

```
files = input("Enter the file name:")
f = open(files,"r")
text = f.read()
print("The words in the file",files,"are",text)
f.close()
print("The",files,"conatains",len(text.split()),"words")
```

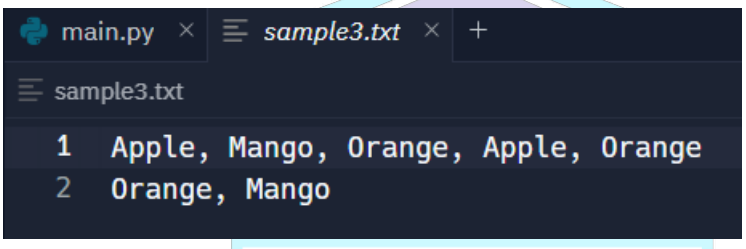
OUTPUT:

CASE -1:




```
Enter File Name: sample1
The sample1.txt contains 1 words
```

CASE - 2:




```
Enter File Name: sample3
The sample3.txt contains 7 words
```

RESULT:

Thus, python program to Count the number of words in a file. has been developed, executed and tested successfully.

Ex. No:		LONGEST WORD IN THE FILE
Date:		

AIM:

To develop a python program to find longest word in the file

ALGORITHM:

Step 1: Start Process.
Step 2: Read file name from user input as file_name.
Step 3: Open file_name as file.
Step 4: Read data from file as file_data and split as words and store in words.
Step 5: Assign variable max_word_list as empty list.
Step 5: If the file is empty, display "the file is empty", else go to step 7.
Step 6: Use for loop to count the maximum words and update max_word_list.
Step 7: Display the max_word_list using for loop.
Step 8: Close file.
Step 9: Stop Process.

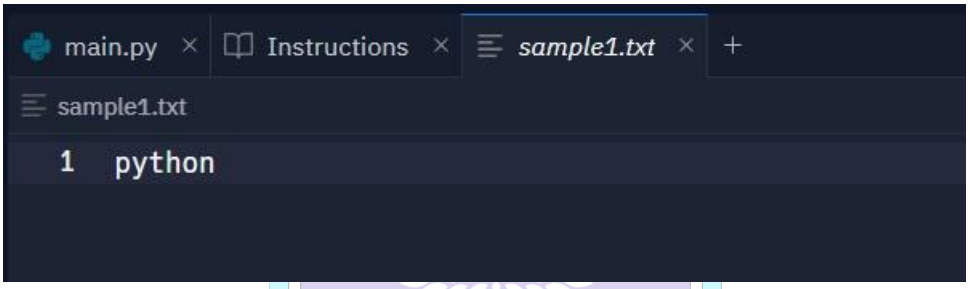
IMPLEMENTATION CODE:

```
file_name = input("Enter File Name: ")
file = open(file_name + ".txt", "r")
file_data = file.read()
words = file_data.split()
max_word_list = []
if words != []:
  max_word = words[0]
  for word in words:
    if len(max_word) <= len(word):
      max_word_list.append(word)
    else:
      if max_word in max_word_list:
        pass
      else:
```

```
            max_word_list.append(max_word)
    print("The longest word in", file_name + ".txt",
"is")
    for word in max_word_list:
      print(word)
else:
    print("The file", file_name + ".txt", "is EMPTY")
file.close()
```

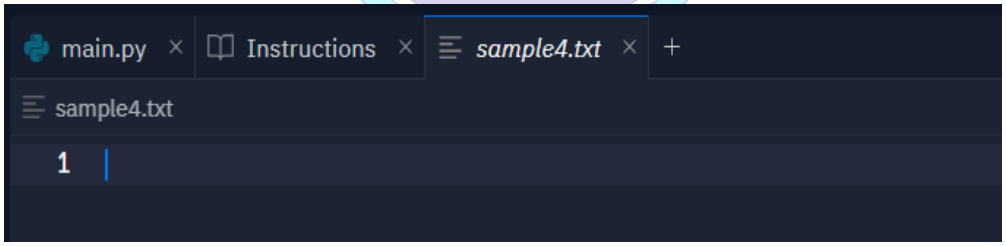
OUTPUT:

CASE – 1:




```
Enter File Name: sample1
The longest word in sample1.txt is
python
```

CASE -2:




```
Enter File Name: sample4
The file sample4.txt is EMPTY
```

RESULT:

Thus, python program to find longest word in the file has been developed, executed and tested successfully.

Ex. No:		PYGAME BASICS
Date:		

AIM:

To develop a python program to draws a circle with pygame.

ALGORITHM:

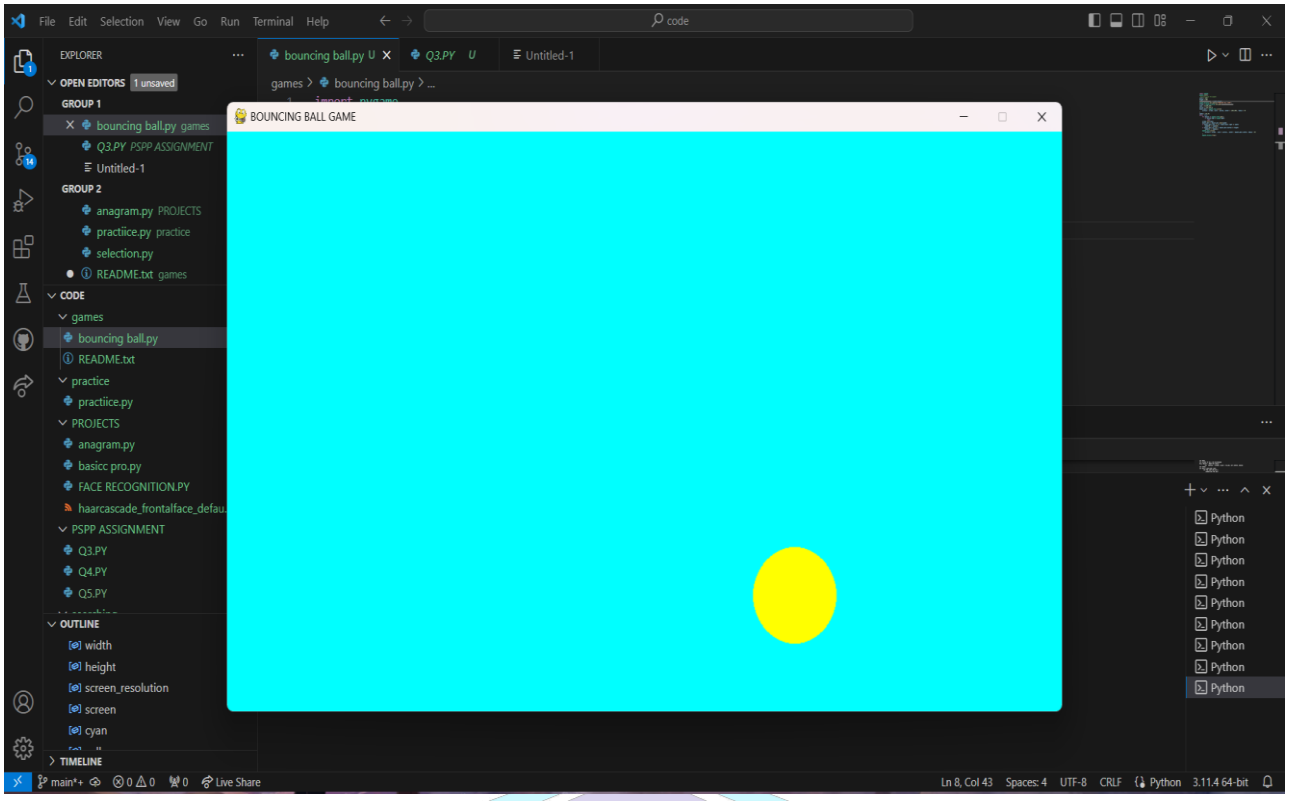
Step 1: Start Process
Step 2: Import pygame module
Step 3: Assign width and height values for the graphical window
Step 4: Assign run as true
Step 5: While run is equal to true do step-6 else goto step-
Step 6: If the user clicks the close button then assign run as False
Step 7: Create the screen object of specific dimension of width and height.
Step 8: Then set the background colour as white
Step 9: Then draw a circle with background color as red and radius as 100
Step 10: Stop process

IMPLEMENTATION CODE:

```
import pygame
pygame.init()
 #initializing the pygame
width = 1000
height = 600
screen_resolution = (width,height)
pygame.display.set_caption("BOUNCING BALL GAME")
screen = pygame.display.set_mode(screen_resolution)
cyan = (0,255,255)
yellow = (255,255,0)
object_ball = pygame.draw.circle(
    surface = screen, color = yellow, center = [100,100], radius = 50
)
speed = [10,10]
while  True:
    for bounce in pygame.event.get():
        if bounce.type == pygame.QUIT:
            pass
    screen.fill(cyan)
    object_ball = object_ball.move(speed)
    if object_ball.left <= 0 or object_ball.right >= width:
```

```
        speed[0] = -speed[0]
    if object_ball.top <=0 or object_ball.bottom >= height:
        speed[1] = -speed[1]
    pygame.draw.circle(
        surface = screen , color = yellow , center = object_ball.center, radius =
50
    )
    pygame.display.flip()
```

OUTPUT:



RESULT:

Thus, python program to draws a circle with pygame has been developed, executed and tested successfully.

Ex.NO: 8(a)	STUDENTS MARK SYSTEM USING NUMPY AND PANDAS

AIM:

To write a Python program to print the student mark system using numpy and panda library.

ALGORITHM:

STEP 1:Create a dictionary 'd' with RollNumber, StudentName, and Score as keys and respective values.

STEP 2:Convert the dictionary 'd' to a dataframe 'df' using pandas.

STEP 3:Add a column 'Grade' to the dataframe 'df' using np.where() and assign the appropriate grade for each student based on their score.

STEP 4:Convert the dataframe 'df' to a string and remove the row indices using the to_string() method.

STEP 5:Print the formatted output.

STEP 6:Generate a bar graph using the Matplotlib library with Student Names on the X-axis and Score on the Y-axis.

STEP 7:Display the graph using plt.show()

PROGRAM:

```
import pandas as pd
import numpy as np


d = {
   'StudentName': [‘Abishek’, ‘Mustafa’ ‘Digeesh’, ‘Arish’],
   'Maths': [90, 80, 70, 60],
   'Science': [85, 75, 65, 55],
   'English': [80, 70, 60, 50]
}
```

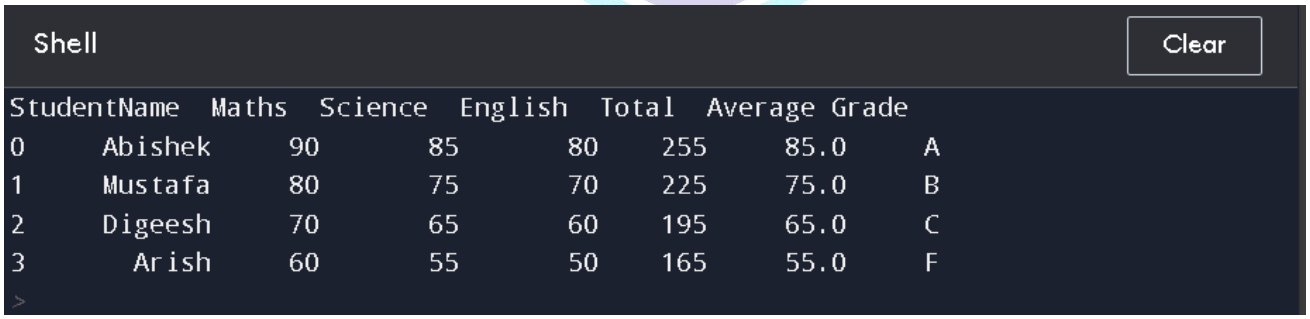
```
df = pd.DataFrame(d)


df['Total'] = df['Maths'] + df['Science'] + df['English']


df['Average'] = df['Total'] / 3
# Add a column for grade
df['Grade'] = np.where(df['Average'] >= 90, 'A+',
                       np.where(df['Average'] >= 80, 'A',
                                np.where(df['Average'] >= 70, 'B',
                                         np.where(df['Average'] >= 60, 'C', 'F'))))


print(df)
```

OUTPUT:



RESULT:

Thus, the above program for printing student mark system using numpy and pandas is executed and the output is verified successfully.