

C.ABDUL HAKEEM COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)

(NBA Accredited & ISO 9001:2000 Certified Institution)

Melvisharam-632 509. Vellore District, Tamil Nadu.

Name.....

Year.....Semester.....Branch.....

Sub.Name:..... Sub.code.....

University Register No.:

CERTIFICATE

Certified that this is the bonafide record of work done by the above student in
the Laboratory during the year 2015-2016.

Signature of Head of the Department

Signature of Lab. Incharge

Submitted for the University Practical Examination held on.....

Examiners

Date :.....

Centre Code :.....

Internal :.....

External :.....

S.NO	DATE	TOPIC	PG.NO	SIGN
1a		Electricity billing		
1b		Retail Shop Billing		
1c		Sin Series		
1d		Compute Electric Current in three phase AC Circuit		
2a		Exchange the values of two variables(without using temp variable)		
2b		Circulate the values of N Variables		
2c		Distance between two points		
3a		Fibonacci Series		
3b		Number Series		
3c		Number Pattern		
3d		Traiangle Pyramid		
4a		Operations of List		
4c		Operations of Tuples		
5a		Operations of Sets		
5b		Operations of Dictionaries		
6a		Factorial using Function		
6b		Largest number in a list using Function		
6c		Area of a Shape using Function		
7a		String Reverse		
7b		Palindrome		
7c		Character Count		
7d		Replacing Characters		
8a		Programs using Written modules and Python Standard Libraries – NUMPY		
8b		Pandas working with Text Data		
8c		Line in a Diagram		
8d		Mark each point with a star		
8e		Solving a set of equations		
9a		Real Time/ Technical Applications using File Handling (copy from one file to another)		

[illegible]

Ex.No: 1a	ELECTRICITY BILLING
Date :	

AIM:

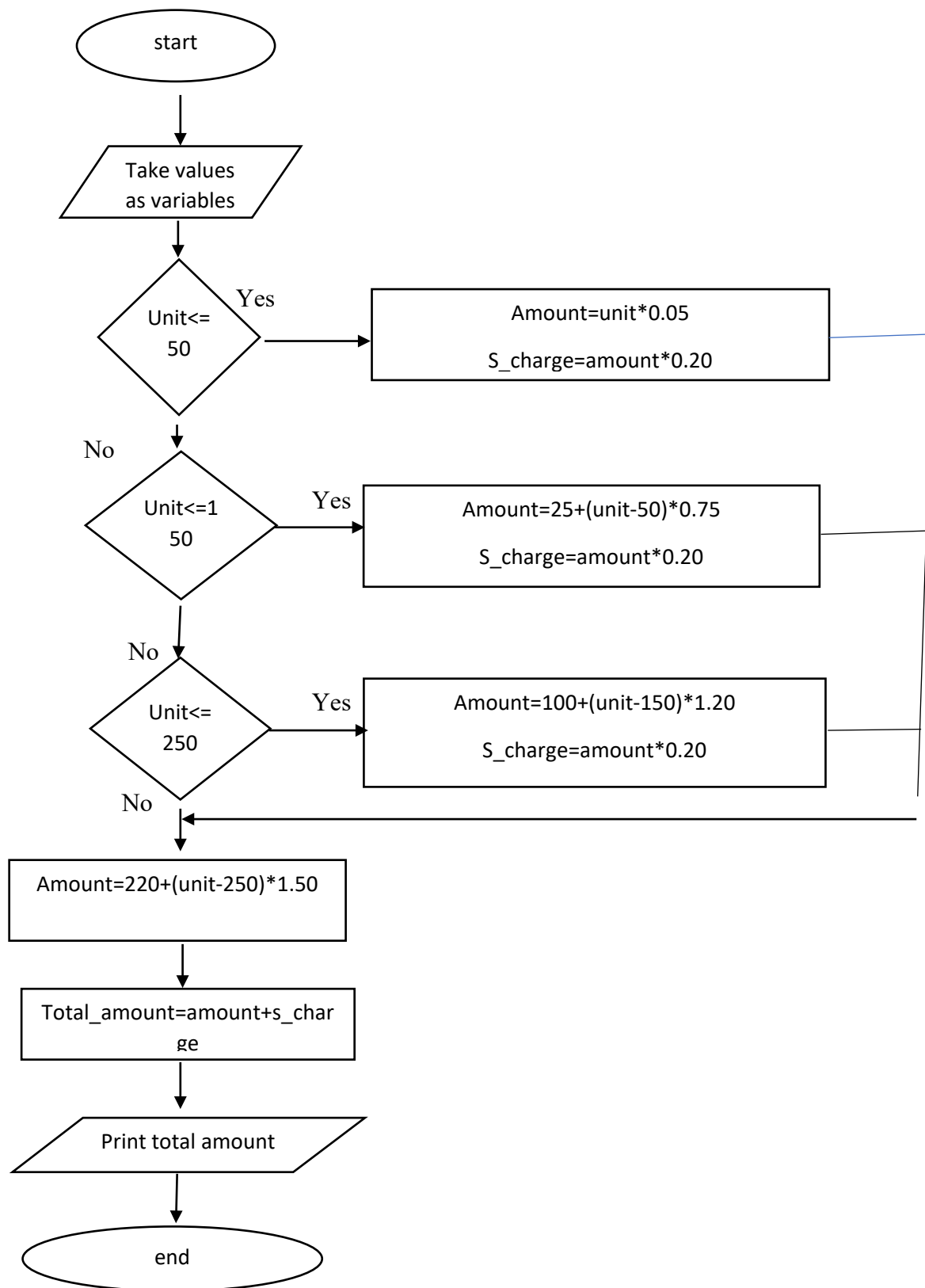
To develop a flow chart to calculate electricity bill.

ALGORITHM:

From Unit To Unit Rate (Rs.) Prices

From Unit	To Unit	Rate (Rs.)	Max.Unit
1	100	0	100
101	200	2	200
201	500	3	500-
-	101 -200	3.5	>500
	201-500	4.6	>500
	>500	606	>500

FLOWCHART:



RESULT:

Thus the flowchart to calculate electricity bill is developed.

Ex.No: 1b	RETAIL SHOP BILLING
Date :	

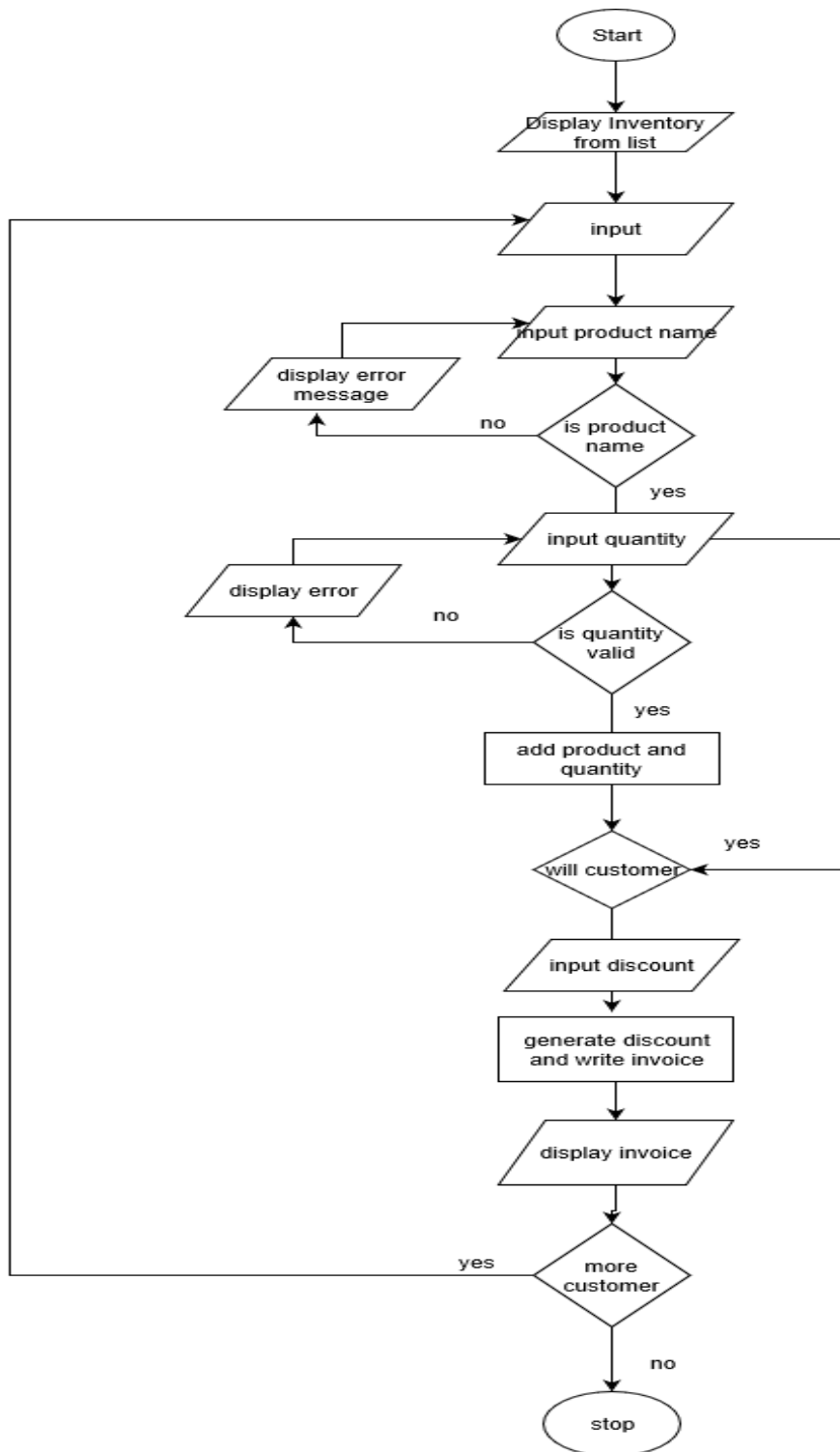
AIM:

To develop a flow chart to calculate retail shop billing.

ALGORITHM:

1. Start the program.
2. Display the products available in the list.
3. Search the product by giving the name, if it is in the list, it checks the quantity otherwise displays error message.
4. Add the product and quantity to invoice.
5. If there is any discount, it generates the discount with invoice.
6. Displays invoice bill.
7. Stop the program.

FLOWCHART:



RESULT:

Thus the flowchart to calculate the retail billing is developed.

Ex.No: 1c	SIN SERIES
Date :	

AIM:

To develop a flowchart to calculate sin series.

ALGORITHM:

1. Start the program.
2. Read the input value.
3. Calculate the TEMP value using the formula $x = x * (3.142/80.0)$.
4. Then the x value is stored in TERM.
5. If the TERM value is greater than epsilon, then we need to calculate the following terms with the formula.

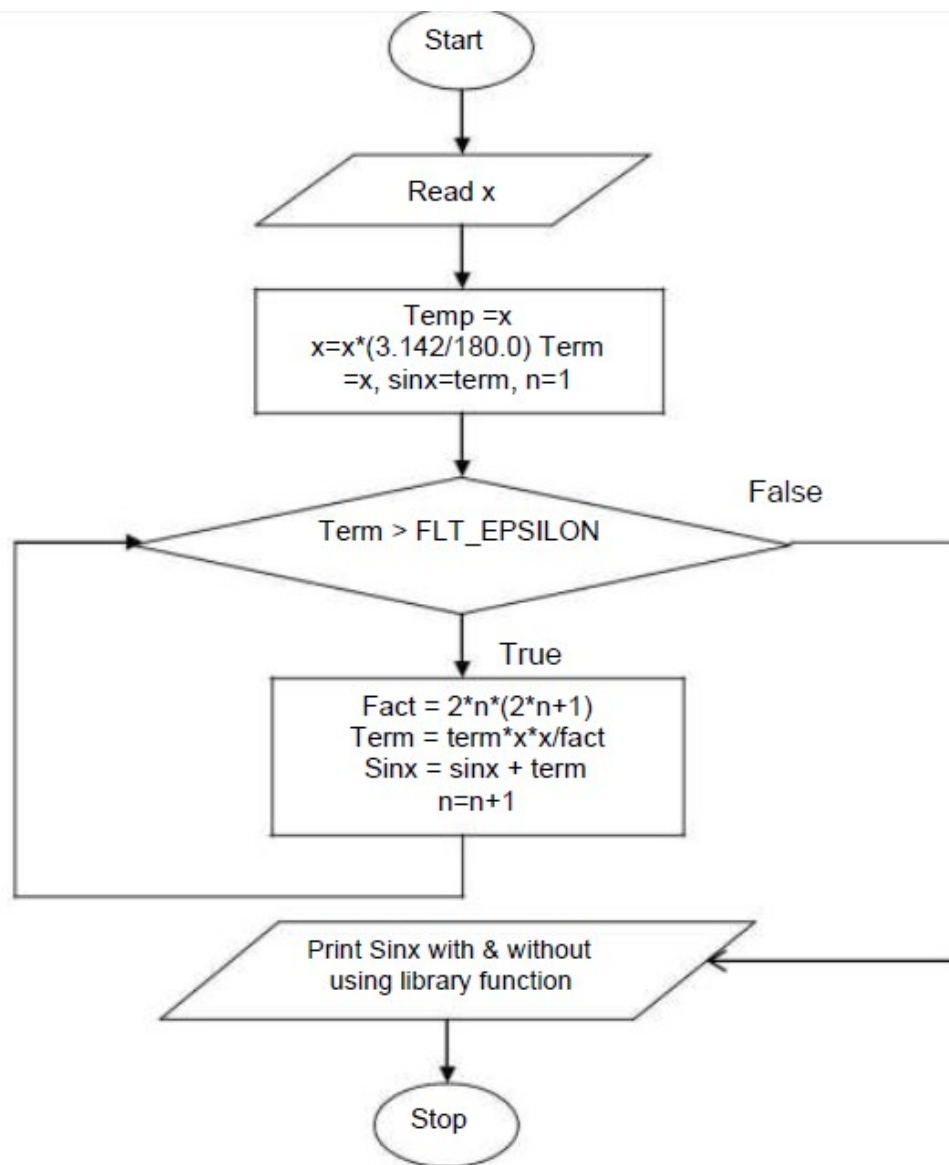
$$\text{Fact} = 2 * n * (2 * n + 1)$$

$$\text{Term} = \text{term} * x * x / \text{fact}$$

$$\text{Sinx} = \text{sinx} + \text{term}$$

$$n = n + 1$$
6. Otherwise, it prints sinx value.
7. Stop the program.

FLOWCHART:



RESULT:

Thus the flowchart to calculate sin series is developed.

Ex.No: 1f	COMUPUTE ELECTRIC CURRENT IN THREE PHASE AC CIRCUIT
Date :	

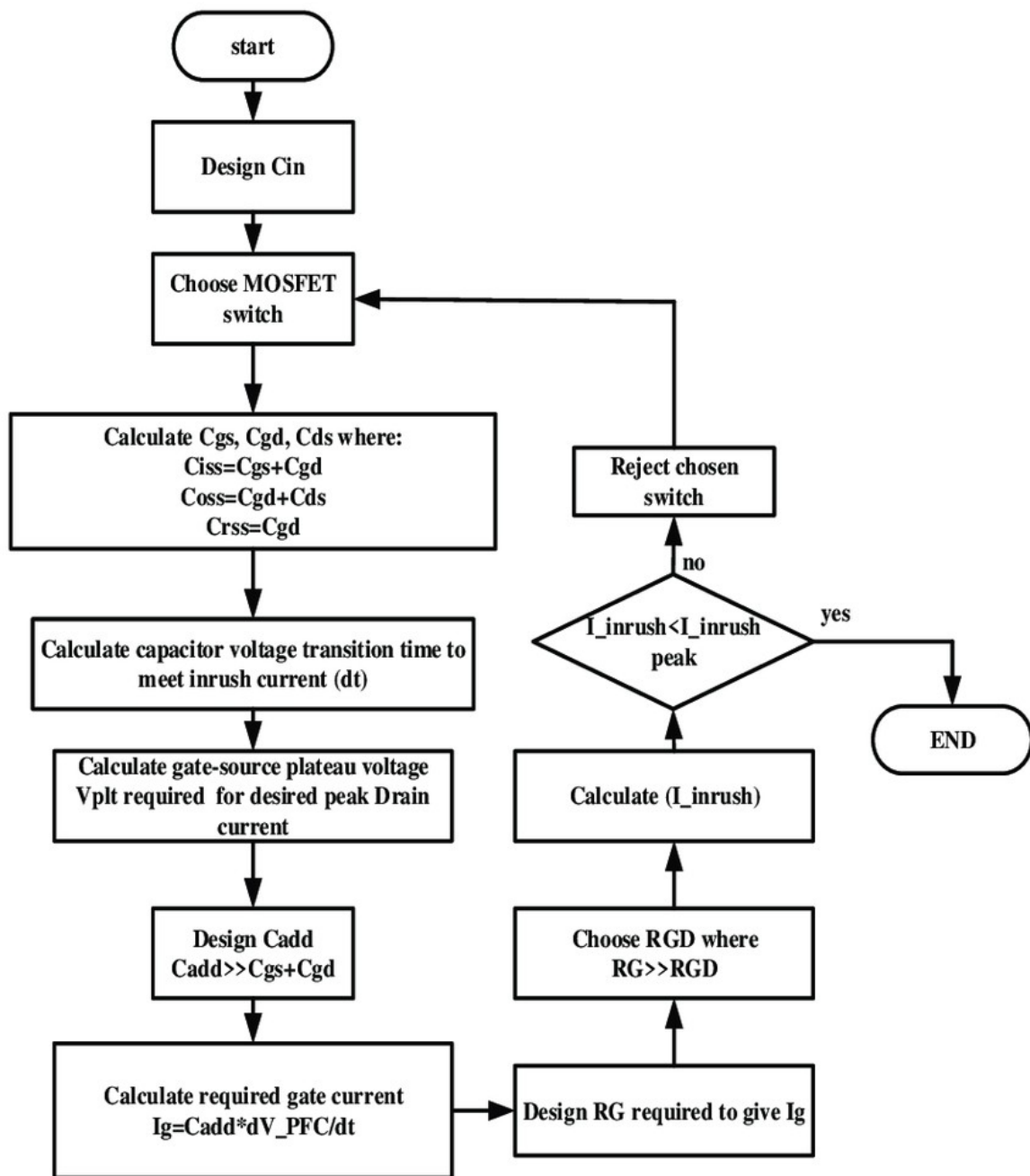
AIM:

To develop a flowchart to compute electric current in three phase AC circuit.

ALGORITHM:

1. Start the program.
2. Design Cin and choose MOSFET switch.
3. Calculate terminal values Cgs (gate-to-source), Cgd (gate-to-drain) and Cds(drain-to-source).
4. Calculate the capacitor voltage transition time to meet inrush current(dt).
5. Calculate gate-source plateau voltage (Vplt).
6. Design Cadd and it should be greater than the values of Cgs and Cgd.
7. Calculate the I_inrush value.
8. The I_inrush value should be less than I_inrush peak value, otherwise the chosen switch is rejected.
9. Stop the program.

FLOWCHART:



RESULT:

Thus the flowchart to compute electric current in three phase AC Circuit is developed.

Ex.No: 2a	EXCHANGE THE VALUES OF TWO VARIABLES (without using temp variable)
Date :	

AIM:

To write a python program to exchange the values of two variables without using temp variable.

ALGORITHM:

1. Start the program.
2. Enter the value of first and second variable.
3. Calculate the values of a and b using the formula
 $a=a+b$
 $b=a-b$
 $a=a-b$
4. Then the values of a and b are get exchanged.
5. Stop the program.

PROGRAM:

```
a=int(input("Enter value of first variable: "))  
b=int(input("Enter value of second variable: "))  
a=a+b  
b=a-b  
a=a-b  
print("a is:",a," b is:",b)
```

OUTPUT:

Enter value of first variable: 2

Enter value of second variable: 4

a is: 4 b is: 2

RESULT:

Thus the program to exchange the values of two variables without using temp variable is executed.

Ex.No: 2b	CIRCULATE THE VALUES OF N VARIABLES
Date :	

AIM:

To write the python program to circulate the values of N variables.

ALGORITHM:

1. Start the program.
2. Define the function to circulate the variables.
3. Assign the given values in a list.
4. Get the value from the user for the number of rotation.
5. Using the slicing concept rotate the list.
6. Assign the variable for the new list after slicing.
7. Print the variable.
8. Stop the program.

PROGRAM:

```
def circulate():  
    l=[10,20,30,40,50,60]  
    n=int(input())  
    output=l[n:]+l[:n]  
    print("After circulating the values are:",output)
```

OUTPUT:

input: 2

After circulating the values are: 30,40,50,60,10,20

input: 4

After circulating the values are: 50,60,10,20,30,40

RESULT:

Thus the python program to circulate the values of N variables is executed.

Ex.No: 2c	DISTANCE BETWEEN TWO POINTS
Date :	

AIM:

To write a python program to calculate distance between two points.

ALGORITHM:

1. Start the program.
2. Enter the values of first coordinate and second coordinate.
3. Values of a and b get splitted by using split() and stored it in p1 and p2.
4. Distance between a and b is calculated by following formula

$$\text{math.sqrt(((int(p1[0])-int(p2[0]))**2)+((int(p1[1])-int(p2[1]))**2))}$$
5. Stop the program.

PROGRAM:

```
import math
a=input("enter first coordinate : ") p1
= a.split(",")
b=input("enter second coordinate : ") p2
= b.split(",")
distance = math.sqrt( ((int(p1[0])-int(p2[0]))**2)+((int(p1[1])-int(p2[1]))**2) )
print("distance between ", a,"and", b, "is",distance)
```

OUTPUT:

enter first coordinate : 3,2

enter second coordinate : 2,6

distance between 3,2 and 2,6 is 4.123105625617661

RESULT:

Thus the python program to calculate distance between two points is executed.

Ex.No: 3a	FIBONACCI SERIES
Date :	

AIM:

To write a python program to calculate Fibonacci series.

ALGORITHM:

1. Start the program.
2. Take an integer variable f1,f2 and f3.
3. Set f1=0 and f2=1.
4. Display f1 and f2.
5. $f3=f1+f2$.
6. Display f3.
7. Start $f1=f2$, $f2=f3$.
8. Repeat 5 -7, for n times.
9. Stop the program.

PROGRAM:

```
n=int(input("enter num"))  
f1=0  
f2=1  
for i in range(0,n):  
    f3=f1+f2  
    print(f1)  
    f1=f2  
    f2=f3
```

OUTPUT:

enter num5

0

1

1

2

3

RESULT:

Thus the program to calculate Fibonacci series is executed.

Ex.No: 3b	NUMBER SERIES
Date:	(1,2,8,16,32...n)

AIM:

To write a python program to calculate the number series.

ALGORITHM:

1. Start the program.
2. Set i as 1.
3. Set n value as greater than i.
4. Display the series as multiplied by two.
5. Stop the program.

PROGRAM:

```
n=int(input("Enter the range of number(Limit):"))
i=1
while i<=n:
    print (i)
    i*=2
```

OUTPUT:

Enter the range of number(Limit):5

1
2
4

RESULT:

Thus the python program to calculate number series is executed.

Ex.No: 3c	NUMBER PATTERN
Date:	

AIM:

To write a python program to display the number pattern.

ALGORITHM:

1. Start the program.
2. Give the number of terms as user input using `int(input())` and store it in a variable.
3. Set val as 1.
4. Loop the number of terms and get incremented as squares and cubes.
5. Print the squares and cubes of numbers in pattern.
6. Stop the program.

PROGRAM:

```
Num_of_terms = int(input("Enter number of terms : "))
val = 1
print ("Number Patterns...")
print ("\nSquares of Numbers : ")
for i in range(1,Num_of_terms+1,1):
    print (i*i)
print "\nCubes of Numbers : ",
for i in range(1,Num_of_terms+1,1):
    print (i*i*i)
```

OUTPUT:

```
Enter number of terms :5
Number Patterns ...
Squares of Numbers : 1 4 9 16 25
Cubes of Numbers :1 8 27 64 125
```

RESULT:

Thus the python program to display the number pattern is executed.

Ex.No: 3d	TRIANGLE PYRAMID
Date:	

AIM:

To write a python program to display the triangle pyramid.

ALGORITHM:

1. Start the program.
2. Give the number of terms as user input using `int(input())` and store it in a variable.
3. Loop from 1 to the number of rows using for loop.
4. Loop from 1 to the first loop interior value using another nested for loop.
5. Inside the inner for loop print the space character.
6. Loop from parent loop iterator value to the number of rows using another for loop(inner for loop).
7. Print the iterator value of the inner for loop.
8. Print the newline character after the end of the inner for loop.
9. Stop the program.

PROGRAM:

```
def triangle(n):
    k = n - 1
    for i in range(0, n):
        for j in range(0, k):
            print(end=" ")
            k = k - 1

        # inner loop to handle number of columns
        # values changing acc. to outer loop
        for j in range(0, i+1):

            # printing stars
            print("* ", end="")

        # ending line after each row
        print("\r")
```

OUTPUT:

```
n = 5
triangle(n)
*
* *
* * *
* * * *
* * * * *
```

RESULT:

Thus the python program to display the triangle pyramid is executed.

Ex.No: 4a	OPERATIONS OF LIST
Date:	

AIM:

To write a python program to evaluate the operations of list.

ALGORITHM:

1. Start the program.
2. Declare the list of items in library.
3. Prints first and fourth element in list.
4. Prints list elements from 0th index to 4th index.
5. Prints -7th or 3rd element from the list.
6. Appending an element from the list using append() function.
7. Finding index of the specified element.
8. Sorting the elements in list.
9. Popping an element using pop() function.
10. Removing the specified element in the list.
11. Inserting an element at specified index.
12. Displays all the elements in the list.
13. Stop the program.

PROGRAM:

```
library=['Books','Periodicals','Newspaper','Manuscripts','Maps','Prints',
'Documents','Ebooks']
print('Library: ',library) print('first
element:          ',library[0])
print('fourth element:      ',library[3])
print('Items in library from 0 to 4 index:      ',library[0:5])
print('3rd or -7th element:      ',library[-7])
print('Library list after append():      ',library)
print('Index of \'newspaper\':      ',library.index('newspaper'))
library.sort()
print('after sorting:      ',library);
print('popped element is:      ',library.pop())
print('after pop(): ',library)
library.remove('maps')
print('after removing \'maps\':      ',library)
library.insert(2,'CDs')
print('after insert:      ',library)
print('Number of elements in Library list:      ',library.count('Ebooks'))
```

OUTPUT:

Library: ['Books', 'Periodicals', 'Newspaper', 'Manuscripts', 'Maps', 'Prints', 'Documents', 'Ebooks']

first element: Books

fourth element: Manuscripts

Items in library from 0 to 4 index: ['Books', 'Periodicals', 'Newspaper', 'Manuscripts', 'Maps']

3rd or -7th element: Periodicals

Library list after append(): ['Books', 'Periodicals', 'Newspaper', 'Manuscripts', 'Maps', 'Prints', 'Documents', 'Ebooks']

RESULT:

Thus the program to evaluate the operations of list is executed.

Ex.No: 4b	OPERATIONS OF TUPLE
Date :	

AIM:

To write a python program to evaluate the operations of tuple.

ALGORITHM:

1. Start the program.
2. Prints the elements in tuple.
3. Prints the first and fourth element of a tuple.
4. Prints tuple elements from 0th index to 4th index.
5. Prints -7th or 3rd element from the tuple.
6. Finding index of the specified element.
7. Displays all the elements in the tuple.
8. Stop the program.

PROGRAM:

```
car = ('Engine','Battery','Alternator','Radiator','Steering','Break','Seat Belt')
print('Components of a car:      ',car)
print('first element:      ',car[0])
print('fourth element:      ',car[3])
print('Components of a car from 0to 4 index:      ',car[0:5])
print('3rd or -7th element: ',car[-7])
print('index of\'Alternator\':      ',car.index('Alternator'))
print('Number of Elements in a car tuple:  ',car.count('Seat Belt'))
print('Length of elements in car tuple:      ',len(car))
```

OUTPUT:

```
Components of a car: ('Engine', 'Battery', 'Alternator', 'Radiator', 'Steering', 'Break', 'Seat Belt')
first element:  Engine
fourth element:      Radiator
Components of a car from 0to 4 index:      ('Engine', 'Battery', 'Alternator', 'Radiator', 'Steering')
3rd or -7th element: Engine
index of'Alternator':  2
Number of Elements in a car tuple:  1
Length of elements in car tuple:      7
```

RESULT:

Thus the program to evaluate the operations of tuple is executed.

Ex.No: 5a	OPERATIONS OF SETS
Date :	

AIM:

To write a python program to evaluate the operations of sets.

ALGORITHM:

1. Start the program.
2. Creating two sets as L1 and L2.
3. Union function is performed for L1 and L2.
4. Intersection is performed for L1 and L2.
5. Differences and symmetric differences of L1 and L2 are performed.
6. Stop the program.

PROGRAM:

```
L1 = {'Pitch', 'Syllabus', 'Script', 'Grammar', 'Sentences'};
L2 = {'Grammar', 'Syllabus', 'Context', 'Words', 'Phonetics'};
print("Union of L1 and L2 is    ",L1|L2)
print("Intersection of L1 and L2 is    ",L1&L2)
print("Differences of L1 and L2 is    ",L1-L2)
print("Symmetric differences of L1 and L2 is    ",L1^L2)
```

OUTPUT:

Union of L1 and L2 is {'Context', 'Grammar', 'Sentences', 'Syllabus', 'Pitch', 'Phonetics', 'Script', 'Words'}

Intersection of L1 and L2 is {'Syllabus', 'Grammar'}

Differences of L1 and L2 is {'Pitch', 'Sentences', 'Script'}

Symmetric differences of L1 and L2 is {'Context', 'Sentences', 'Pitch', 'Script', 'Words', 'Phonetics'}

RESULT:

Thus the python program to evaluate the operations of sets is executed.

Ex.No: 5b	OPERATIONS OF DICTIONARY
Date :	

AIM:

To write a python program to evaluate the operations of dictionary.

ALGORITHM:

1. Start the program.
2. Create items in dictionary
3. Create items in dictionary.
4. Prints keys and values in dictionary.
5. Updates the values in dictionary.
6. Deleting the value from the dictionary.
7. Using get method to access dictionary.
8. Using pop method to delete element in the dictionary.
9. Using has_key method to delete element in dictionary.
10. Checks whether the particular key is present in the dictionary or not.
11. Stop the program.

PROGRAM:

```
dict={'chassis':{1:'frames',2:'suspension system',3:'wheels', 4:'axles'},'engine':{1:'internal
combustion engine',2:'ignition engine'},'transmission
system':{1:'clutch',2:'gearbox',3:'propeller shaft'}}
print("\n\nDictionary key and values")
print(dict.keys())
print(dict.values())
print("\n\nAccessing Dictionary- print the attribute from nested dictionary")
print(dict['chassis'][1])
print("\n\nUpdating the values in dictionary")
dict['chassis'][4]='live axles'
print(dict.values())
print("\n\ndeleting from a dictionary")
del [dict['transmission system'][3]]
print(dict.values())
print("\n\nAdding values into dictionary")
dict['transmission system'][4]='Differential'
print(dict.values())
print("\n\nUsing get method to access dictionary")
print(dict.get('chassis'))
print("\n\nusing pop method to delete element in dictionary")
dict.pop('engine')
print(dict.values())
print("\n\nusing has_key method to check dictionary keys")
print("Is chassis is key present in dictionary?")
print(dict.has_key('chassis'))
print("Is engine is key present in dictionary?")
print(dict.has_key('engine'))
```

OUTPUT:

Dictionary key and values

```
dict_keys(['chassis', 'engine', 'transmission system'])
```

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'axles'}, {1: 'internal combustion
engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox', 3: 'propeller shaft'} ])
```

Accessing Dictionary- print the attribute from nested dictionary

```
frames
```

Updating the values in dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'internal
combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox', 3: 'propeller shaft'} ])
```

deleting from a dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'internal
combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox'} ])
```

Adding values into dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'internal
combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox', 4: 'Differential'} ])
```

Using get method to access dictionary

```
{1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}
```

using pop method to delete element in dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'clutch', 2: 'gearbox', 4: 'Differential'} ])
```

using has_key method to check dictionary keys

Dictionary key and values

```
dict_keys(['chassis', 'engine', 'transmission system'])
```

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'axles'}, {1: 'internal combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox', 3: 'propeller shaft'} ])
```

Accessing Dictionary- print the attribute from nested dictionary

frames

Updating the values in dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'internal combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox', 3: 'propeller shaft'} ])
```

deleting from a dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'internal combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox'} ])
```

Adding values into dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'internal combustion engine', 2: 'ignition engine'}, {1: 'clutch', 2: 'gearbox', 4: 'Differential'} ])
```

Using get method to access dictionary

```
{1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}
```

using pop method to delete element in dictionary

```
dict_values([ {1: 'frames', 2: 'suspension system', 3: 'wheels', 4: 'live axles'}, {1: 'clutch', 2: 'gearbox', 4: 'Differential'} ])
```

using has_key method to check dictionary keys

Is chassis key present in dictionary?

RESULT:

Thus the python program to evaluate the operations of dictionary is evaluated.

Ex.No: 6a	FACTORIAL USING FUNCTION
Date :	

AIM:

To write a python program to evaluate the factorial using function.

ALGORITHM:

1. Start the program.
2. Enter the number to calculate the factorial value.
3. If the given value is less than zero, it prints as factorial does not exist for negative numbers.
4. If the given number is 0, the factorial of 0 is 1.
5. If the given number is greater than 0, for loop is executed for the factorial calculation.
factorial=factorial*i
6. Stop the program.

PROGRAM:

```
def factorial(num):  
    factorial=1  
    if num < 0:  
        print("sorry, factorial does not exist for negative numbers")  
    elif num == 0:  
        print("The factorial of 0 is 1")  
    else:  
        for i in range(1,num+1):  
            factorial=factorial*i  
    print 'The factorial of', num ,'is' , factorial  
factorial(num=int(input('Enter number:')))
```

OUTPUT:

Enter number: 6

The factorial of 6 is 720

RESULT:

Thus the python program to evaluate the factorial using function is executed.

Ex.No: 6b	LARGEST NUMBER IN A LIST USING FUNCTION
Date :	

AIM:

To write a python program to calculate largest number in a list using function.

ALGORITHM:

1. Start the program.
2. Creating an empty list.
3. Enter number of elements in list as user input.
4. Iterate the loop for each element in the list.
5. Prints the maximum element.
6. Stop the program.

PROGRAM:

```
def my_large_elem(num):  
    list1 = []  
  
    for i in range(1, num + 1):  
        ele = int(input("Enter a element: "))  
        list1.append(ele)  
    print "Largest element is:", max(list1)  
my_large_elem(num = int(input("Enter number of elements in list: ")))
```

OUTPUT:

```
Enter number of elements in list: 3  
Enter a element: 12  
Enter a element: 122  
Enter a element: 2333  
Largest element is: 2333
```

RESULT:

Thus the program to calculate the largest number in a list using function is executed.

Ex.No: 6c	AREA OF SHAPE USING FUNCTION
Date :	

AIM:

To write a python program to calculate the area of shape using function.

ALGORITHM:

1. Start the program.
2. Calculate the area by giving the name of the shape.
3. If it is rectangle, enter the value of length and breadth.
4. Area of rectangle is calculated by using the formula
rect_area=length*breadth
5. If it is square, enter the side length.
6. Area of square is calculated by using the formula
sqt_area=s*s
7. If it is triangle, enter triangle's height and breadth length.
8. Area of triangle is calculated by using the formula
tri_area=0.5*b*h
9. If it is circle, enter circle's radius length.
10. Area of circle is calculated by using the formula
circ_area = pi*r*r
11. If it is parallelogram, enter the base and height length.
12. Area of parallelogram is calculated by using the formula
para_area=b*h
13. Stop the program.

PROGRAM:

```
def calculate_area(name):\n\n    # converting all characters\n    # into lower cases\n    name = name.lower()\n\n    # check for the conditions\n    if name == "rectangle":\n        l = int(input("Enter rectangle's length: "))\n        b = int(input("Enter rectangle's breadth: "))\n\n        # calculate area of rectangle\n        rect_area = l * b\n        print(f"The area of rectangle is\n            {rect_area}.")\n\n    elif name == "square":\n        s = int(input("Enter square's side length: "))\n\n        # calculate area of square\n        sqt_area = s * s\n        print(f"The area of square is\n            {sqt_area}.")\n\n    elif name == "triangle":\n        h = int(input("Enter triangle's height length: "))\n        b = int(input("Enter triangle's breadth length: "))\n\n        # calculate area of triangle\n        tri_area = 0.5 * b * h\n        print(f"The area of triangle is\n            {tri_area}.")\n\n    elif name == "circle":\n        r = int(input("Enter circle's radius length: "))\n        pi = 3.14\n\n        # calculate area of circle\n        circ_area = pi * r * r\n        print(f"The area of triangle is\n            {circ_area}.")\n\n    elif name == 'parallelogram':\n        b = int(input("Enter parallelogram's base length: "))\n        h = int(input("Enter parallelogram's height length: "))
```



```

# calculate area of parallelogram
para_area = b * h
print(f"The area of parallelogram is
      {para_area}.")

else:
    print("Sorry! This shape is not available")

# driver code
if __name__ == "__main__" :

    print("Calculate Shape Area")
    shape_name = input("Enter the name of shape whose area you want to find: ")

    # function calling
    calculate_area(shape_name)

```

OUTPUT:

Calculate Shape Area

Enter the name of shape whose area you want to find: rectangle

Enter rectangle's length: 3

Enter rectangle's breadth: 3

The area of rectangle is 9.

Calculate Shape Area

Enter the name of shape whose area you want to find: square

Enter square's side length: 4

The area of square is 16.

Calculate Shape Area

Enter the name of shape whose area you want to find: parallelogram

Enter parallelogram's base length: 5

Enter parallelogram's height length: 6

The area of parallelogram is 30.

Calculate Shape Area

Enter the name of shape whose area you want to find: circle

Enter circle's radius length: 7

The area of circle is 153.86.

Calculate Shape Area

Enter the name of shape whose area you want to find: triangle

Enter triangle's height length: 8

Enter triangle's breadth length: 9

The area of triangle is 36.0.

RESULT:

Thus the python program to calculate the area of shape using function is executed.

Ex.No: 7a	STRING REVERSE
Date :	

AIM:

To write a python program to calculate the reverse of a string.

ALGORITHM:

1. Start the program.
2. A method named reverse is defined, that takes a string as a parameter.
3. It checks the length of the string and if it is not 0, then the function is called again on all elements except the first element of the string, and the first element of the string is concatenated to the result of this function call.
4. By default, the string name is given as input in variable s.
5. Prints the original string and the reversed string.
6. Stop the program.

PROGRAM:

```
def reverse(s):  
    if len(s) == 0:  
        return s  
    else:  
        return reverse(s[1:]) + s[0]
```

```
s = "Computer Science and Engineering"
```

```
print ("The original string is : ",end="")  
print (s)
```

```
print ("The reversed string(using recursion) is : ",end="")  
print (reverse(s))
```

OUTPUT:

The original string is : Computer Science and Engineering

The reversed string(using recursion) is : gnireenignE dna ecneicS retupmoC

RESULT:

Thus the python program to calculate the reverse of a string is executed.

Ex.No: 7b	PALINDROME
Date:	

AIM:

To write a python program for palindrome.

ALGORITHM:

1. Start the program.
2. Read the number or letter.
3. Hold the number or letter in the temporary variable.
4. Reverse of number or letter is generated from the pattern `isPalindrome(s[1:1-1])`.
5. This generated output is compared with the actual value.
6. When the generated value is an exact reverse of the given string, then the output is printed as "Yes! it is a palindrome".
7. Otherwise, "No, it is not a palindrome".
8. Stop the program.

PROGRAM:

```
def isPalindrome(s):
    l = len(s)
    if l < 2:
        return True
    elif s[0] == s[l - 1]:
        print (s)
        return isPalindrome(s[1: l - 1])
    else:
        print ("checking
        fails")
        return False
s = raw_input("enter a string value to check whether it is a palindrome or not: ")
print ("-----This is how it checks!!! ----- ")
ans = isPalindrome(s)
if ans:
    print("=====completed checking=====")
    print("Yes! it is a palindrome")
else:
    print("No! it is not a palindrome")
```

OUTPUT:

```
enter a string value to check whether it is a palindrome or not: yes
-----This is how it checks!!!

checking fails

No! it is not a palindrome

enter a string value to check whether it is a palindrome or not: madam
-----This is how it checks!!!

s
s

=====completed checking=====

Yes! it is a palindrome
```

RESULT:

Thus the python program for palindrome is executed.

Ex.No: 7c	CHARACTER COUNT
Date :	

AIM:

To write a python program to count the character.

ALGORITHM:

1. Start the program.
2. Read the given string and set count value as 0.
3. Counts the string value and gets incremented by 1, whenever it reads the string.
4. Displays the total number of characters present in the given string.
5. Stop the program.

PROGRAM:

```
string= 'Python is the best programming language'  
count = 0
```

```
for i in range(0, len(string)):  
    if (string[i] != ' '):  
        count = count + 1
```

```
# Displays the total number of characters present in the given string
```

```
print("Total number of characters in the given string is: " + str(count))
```

OUTPUT:

Total number of characters in the given string is: 34

RESULT:

Thus the python program to count the character is executed.

Ex.No: 7d	REPLACING CHARACTERS
Date :	

AIM:

To write a python program to replace the characters.

ALGORITHM:

1. Start the program.
2. Read the given text.
3. Some words can replace by using the replace().
4. Prints the replaced text.
5. Likewise, characters can also be replaced by using the replace().
6. Stop the program.

PROGRAM:

```
txt = "I like bananas, bananas are sweet "  
x = txt.replace("bananas", "apples")  
print ("this is string replace")  
print(x)  
# I=> i num of occurence  
x = txt.replace("bananas", "apple", 1) print (x)  
x=txt.replace("a","e")  
print ("this is character replace")  
print(x)  
# I=> i num of occurence x = txt.replace("a", "e", 3) print x
```

OUTPUT:

```
this is string replace  
I like apples, apples are sweet  
I like apple, bananas are sweet  
this is character replace  
I like benenes, benenes ere sweet
```

RESULT:

Thus the python program for replacing the characters is executed.

Ex.No: 8a	PROGRAMS USING WRITTEN MODULES AND PYTHON STANDARD LIBRARIES(numpy)
Date :	

AIM:

To write a python program using written modules and python standard libraries(numpy).

ALGORITHM:

1. Start the program.
2. Import the numpy package.
3. Values are stored in array as arr.
4. First, the initial array gets displayed.
5. Values are get sliced and displayed.
6. Mathematical operations are performed on elements and displayed.
7. Stop the program.

PROGRAM:

```
import numpy as np
arr = np.array([1, 2, 3])
print("Array with Rank 1: \n", arr)
arr = np.array([[1, 2, 3],[4, 5, 6]])
print("Array with Rank 2: \n", arr)
arr = np.array((1, 3, 2))
print("\nArray created using passed tuple:\n", arr)
arr = np.array([[-1, 2, 0, 4],
                [4, -0.5, 6, 0],
                [2.6, 0, 7, 8],
                [3, -7, 4, 2.0]])
print("Initial Array: ")
print(arr)
sliced_arr = arr[:2, ::2]
print ("Array with first 2 rows and alternate columns(0 and 2):\n", sliced_arr)
Index_arr = arr[[1, 1, 0, 3],
                 [3, 2, 1, 0]]
print ("\nElements at indices (1, 3), (1, 2), (0, 1), (3, 0):\n", Index_arr)
a = np.array([[1, 2],
               [3, 4]])
b = np.array([[4, 3],
               [2, 1]])
print ("Adding 1 to every element:", a + 1)
print ("\nSubtracting 2 from each element:", b - 2)
print ("\nSum of all array elements: ", a.sum())
print ("\nArray sum:\n", a + b)
```

OUTPUT:

Array with Rank 1:

```
[1 2 3]
```

Array with Rank 2:

```
[[1 2 3]
```

```
[4 5 6]]
```

Array created using passed tuple:

```
[1 3 2]
```

Initial Array:

```
[[-1.  2.  0.  4. ]
```

```
[ 4. -0.5  6.  0. ]
```

```
[ 2.6  0.  7.  8. ]
```

```
[ 3. -7.  4.  2. ]]
```

Array with first 2 rows and alternate columns(0 and 2):

```
[[-1.  0.]
```

```
[ 4.  6.]]
```

Elements at indices (1, 3), (1, 2), (0, 1), (3, 0):

```
[0.  6.  2.  3.]
```

Adding 1 to every element:

```
[[2 3]
```

```
[4 5]]
```

Subtracting 2 from each element:

```
[[ 2  1]
```

```
[ 0 -1]]
```

Sum of all array elements: 10

Array sum:

```
[[5 5]
```

```
[5 5]]
```

RESULT:

Thus the programs using written modules and python standard libraries(numpy) is executed.

Ex.No: 8b	PANDAS WORKING WITH TEXT DATA
Date :	

AIM:

To write a python program for pandas working with text data.

ALGORITHM:

1. Start the program.
2. Import the pandas package.
3. Define a dictionary containing employee data.
4. Details of employee are framed by using DataFrame().
5. Convert the dictionary into DataFrame.
6. Converting and overwriting the values in column.
7. Stop the program.

PROGRAM:

```
import pandas as pd
# Define a dictionary containing employee data
data = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32],
        'Address':['Delhi', 'Kanpur', 'Allahabad', 'Kannauj'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}
# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# converting and overwriting values in column
df["Name"] = df["Name"].str.lower()
print(df)
```

OUTPUT:

	Name	Age	Address	Qualification
0	Jai	27	Delhi	Msc
1	Princi	24	Kanpur	MA
2	Gaurav	22	Allahabad	MCA
3	Anuj	32	Kannauj	Phd

RESULT:

Thus the python program for pandas working with text data is executed.

Ex.No: 8c	LINE IN A DIAGRAM
Date :	

AIM:

To write a python program to draw line in a diagram.

ALGORITHM:

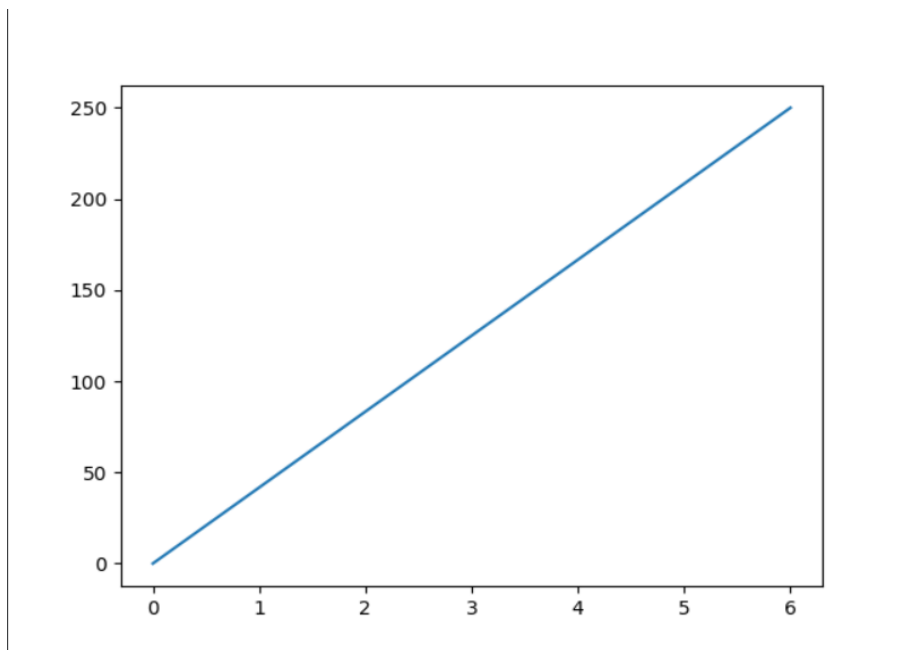
1. Start the program.
2. Import matplotlib and numpy packages.
3. Points are defined in an array.
4. Points are plotted by using plot().
5. Displays the plotted values.
6. Stop the program.

PROGRAM:

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```

OUTPUT:**RESULT:**

Thus the python program to draw line in a diagram is executed.

Ex.No: 8d	MARK EACH POINT WITH A STAR
Date :	

AIM:

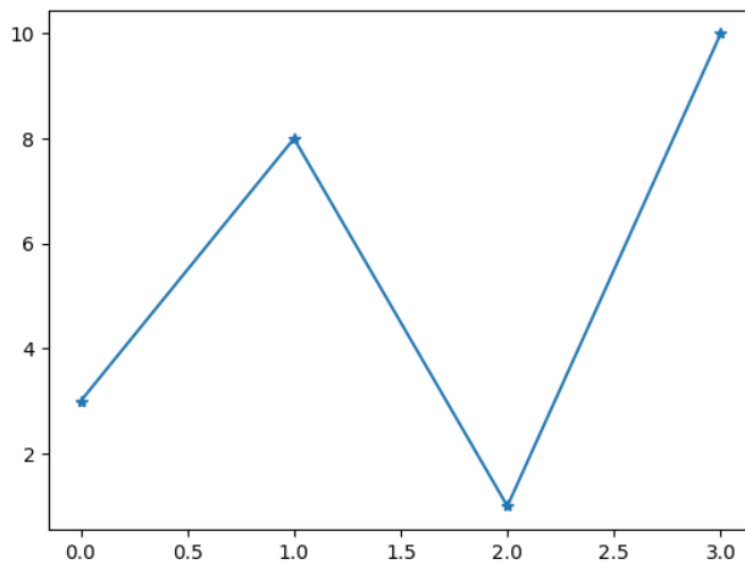
To write a python program to mark each point with a star.

ALGORITHM:

1. Start the program.
2. Import matplotlib and numpy packages.
3. Points are defined in an array.
4. Points are plotted by using plot().
5. Displays the plotted values.
6. Stop the program.

PROGRAM:

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, marker = '*')
plt.show()
```

OUTPUT:**RESULT:**

Thus the python program to mark each point with a star is executed.

Ex.No: 8e	SOLVING A SET OF EQUATIONS
Date :	

AIM:

To write a python program for solving a set of equations.

ALGORITHM:

1. Start the program.
2. Import scipy and numpy packages.
3. Points are defined in an array.
4. Points are solved by using linalg.solve().
5. Displays the solved values.
6. Stop the program.

PROGRAM:

```
from scipy import linalg
import numpy
X=numpy.array([[4,3],[3,4]])
Y=numpy.array([[12],[18]])
print(linalg.solve(X,Y))
X.dot(linalg.solve(X,Y))-Y
```

OUTPUT:

```
[[-0.85714286]
 [ 5.14285714]]
```

RESULT:

Thus the python program for solving a set of equations is executed.

Ex.No: 9a	REAL-TIME/TECHNICAL APPLICATIONS USING FILE HANDLING. (copy from one file to another)
Date :	

AIM:

To write a python program for copying from one file to another file.

ALGORITHM:

1. Start the program.
2. Create first file and second file, save it in the directory.
3. Reads content from the first file.
4. Appends the content to the second file.
5. Import the shutil package.
6. Using copyfile() method, content of first file is copied to the second file.
7. Stop the program.

PROGRAM:

with open('first.txt', 'r') as firstfile, open('second.txt', 'a') as secondfile:

read content from first file

for line in firstfile:

append content to second file

secondfile.write(line)

SAME APPROACH

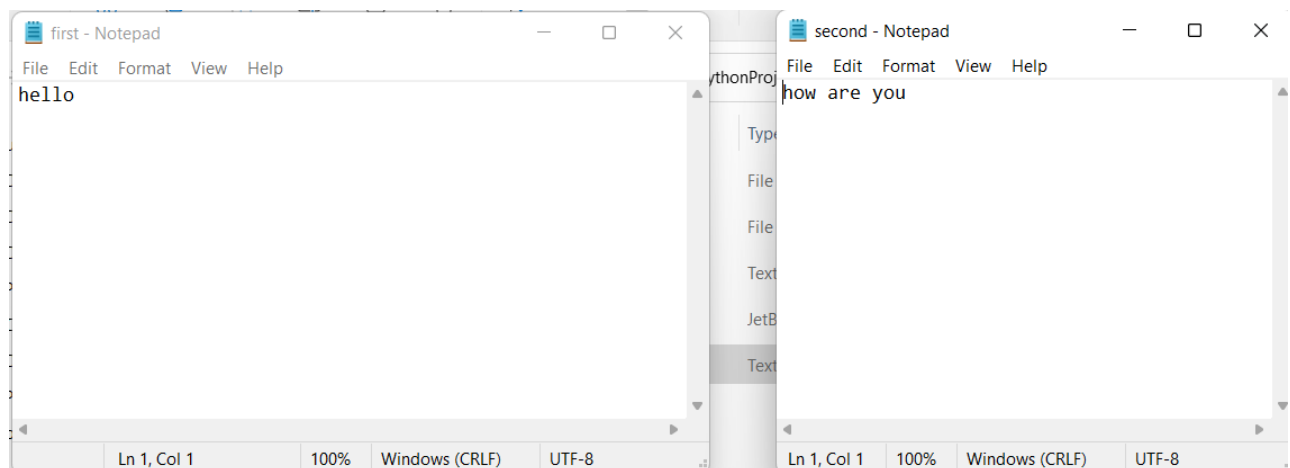
import shutil

use copyfile()

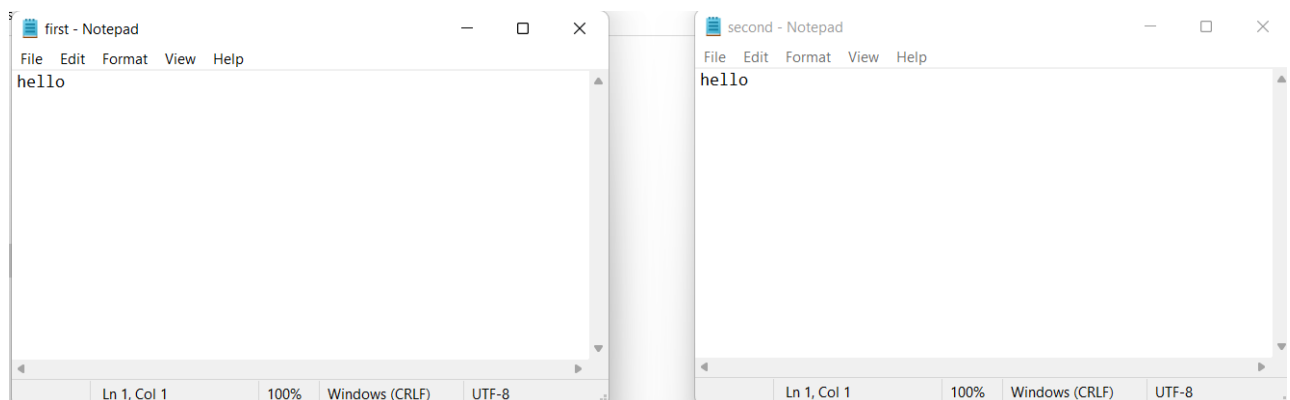
shutil.copyfile('first.txt', 'second.txt')

OUTPUT:

Before copying....



After Copying....



RESULT:

Thus the python program for copying from one file to another file is executed.

Ex.No: 9b	REAL-TIME/TECHNICAL APPLICATIONS USING FILE HANDLING. (word count)
Date :	

AIM:

To write a python program to count the characters in a file.

ALGORITHM:

1. Start the program.
2. Import the sys package.
3. Number of lines, characters and words in the file are initialized to 0.
4. By using the split() function, number of lines, characters and words are counted and displays as result.
5. Stop the program.

PROGRAM:

```
import sys
num_lines=0
num_words=0
num_chars=0
with open('word.txt','r')as f:
    for line in f:
        words=line.split()
        num_lines+=1
        num_words+=len(words)
        num_chars+=len(line)
print(num_lines,num_words,num_chars)
```

OUTPUT:

1 4 19

2 8 51

RESULT:

Thus the python program to count characters in file is executed.

Ex.No: 9c	REAL-TIME/TECHNICAL APPLICATIONS USING FILE HANDLING. (longest word)
Date :	

AIM:

To write a python program to identify the longest word in a file.

ALGORITHM:

1. Start the program.
2. Define s as a variable.
3. Identify the largest word in the variable.
4. By using `list(s.split(" "))`, the largest word is identified and displayed as a output.
5. Stop the program.

PROGRAM:

```
def largestWord(s):  
    s = sorted(s, key = len)  
    print(s[-1])  
if __name__ == "__main__":  
    s = "be confident and be yourself"  
    l = list(s.split(" "))  
    largestWord(l)
```

OUTPUT:

confident

RESULT:

Thus the python program to identify the longest word in a file is executed.

Ex.No: 10a	REAL-TIME/TECHNICAL APPLICATIONS USING EXCEPTION HANDLING. (divide by zero error)
Date :	

AIM:

To write a python program to get the zero division error.

ALGORITHM:

1. Start the program.
2. Read the values as num1 and num2.
3. Division is performed and the result gets displayed.
4. As exception, the ValueError and ZeroDivision Error was found.
5. Stop the program.

PROGRAM:

```
try:
    num1 = int(input("Enter First Number: "))
    num2 = int(input("Enter Second Number: "))
    result = num1 / num2
    print(result)
except ValueError as e:
    print("Invalid Input Please Input Integer...")
except ZeroDivisionError as e:
    print(e)
```

OUTPUT:

Enter First Number: 5

Enter Second Number: 10

0.5

RESULT:

Thus the python program to get zero division error is executed.

Ex.No: 10b	REAL-TIME/TECHNICAL APPLICATIONS USING EXCEPTION HANDLING. (voter's age validity)
Date :	

AIM:

To write a python program to validate the voter's age.

ALGORITHM:

1. Start the program.
2. Age of a particular person is accepted as a user input.
3. If the age is greater than 18, prints as "eligible to vote".
4. Otherwise prints as "not eligible to vote".
5. By exception, "enter a valid age" message gets displayed.
6. Stop the program.

PROGRAM:

```
def main():  
    try:  
        age=int(input("enter your age"))  
        if age>=18:  
            print ("eligible to vote")  
        else:  
            print("not eligible to vote")  
    except:  
        print("enter a valid age")  
main()
```

OUTPUT:

```
enter your age:21  
eligible to vote  
enter your age: 15  
not eligible to vote  
enter a valid age
```

RESULT:

Thus the python program to validate the voter's age is executed.

Ex.No: 10c	REAL-TIME/TECHNICAL APPLICATIONS USING EXCEPTION HANDLING. (display of students mark)
Date :	

AIM:

To write a python program to validate the students mark range.

ALGORITHM:

1. Start the program.
2. Empty array is initialized with the variable marks.
3. Student details such as roll number, name, marks in each subject are entered.
4. It consolidates and displays the total and average marks got by that particular student.
5. Prints the details of student by entering their roll number.
6. Stop the program.

PROGRAM:

```
class Student:
```

```
    marks = []
```

```
    def getData(self, rn, name, m1, m2, m3):
```

```
        Student.rn = rn
```

```
        Student.name = name
```

```
        Student.marks.append(m1)
```

```
        Student.marks.append(m2)
```

```
        Student.marks.append(m3)
```

```
    def displayData(self):
```

```
        print ("Roll Number is: ", Student.rn)
```

```
        print ("Name is: ", Student.name)
```

```
        #print ("Marks in subject 1: ", Student.marks[0])
```

```
        #print ("Marks in subject 2: ", Student.marks[1])
```

```
        #print ("Marks in subject 3: ", Student.marks[2])
```

```
        print ("Marks are: ", Student.marks)
```

```
        print ("Total Marks are: ", self.total())
```

```
        print ("Average Marks are: ", self.average())
```

```
    def total(self):
```

```
        return (Student.marks[0] + Student.marks[1] + Student.marks[2])
```

```
    def average(self):
```

```
        return ((Student.marks[0] + Student.marks[1] + Student.marks[2])/3)
```

```
r = int (input("Enter the roll number: "))
```

```
name = input("Enter the name: ")
```

```
m1 = int (input("Enter the marks in the first subject: "))
```

```
m2 = int (input("Enter the marks in the second subject: "))
```

```
m3 = int (input("Enter the marks in the third subject: "))
```

```
s1 = Student()  
s1.getData(r, name, m1, m2, m3)  
s1.displayData()
```

OUTPUT:

Enter the roll number: 10
Enter the name: karthik
Enter the marks in the first subject: 95
Enter the marks in the second subject: 98
Enter the marks in the third subject: 82
Roll Number is: 10
Name is: karthik
Marks are: [95, 98, 82]
Total Marks are: 275

RESULT:

Thus the python program to validate the students mark range is executed.

Ex.No: 11

Date :

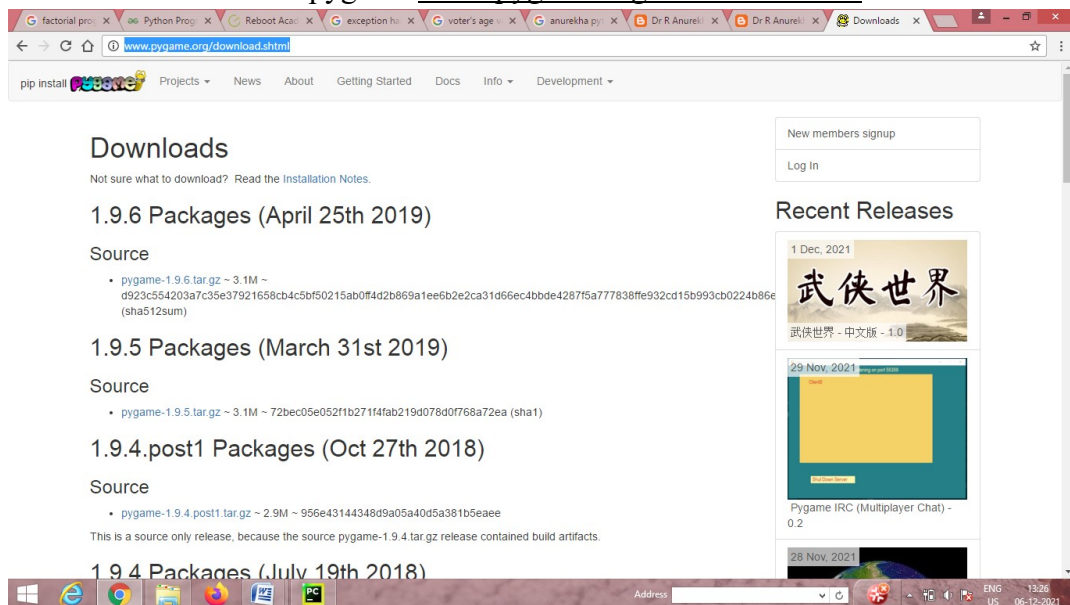
EXPLORING PYGAME TOOL.

AIM:

To explore pygame tool in python

PROCEDURE:

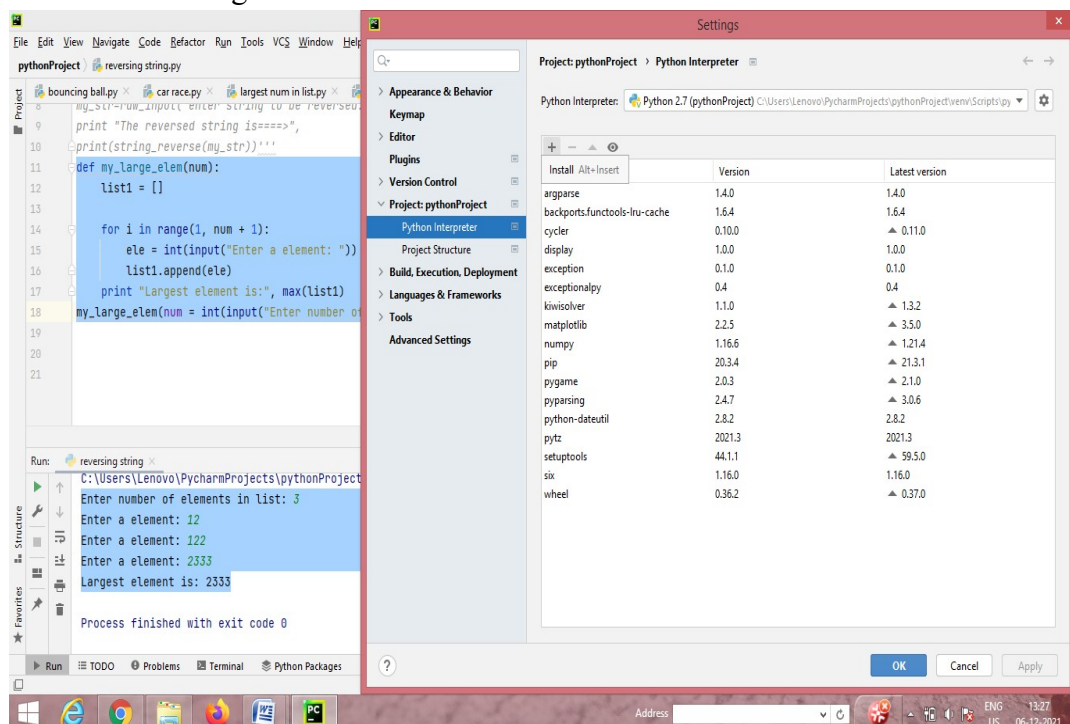
1. Install python 3.6.2 into c:\
2. Go to the link to install pygame www.pygame.org/download.shtml



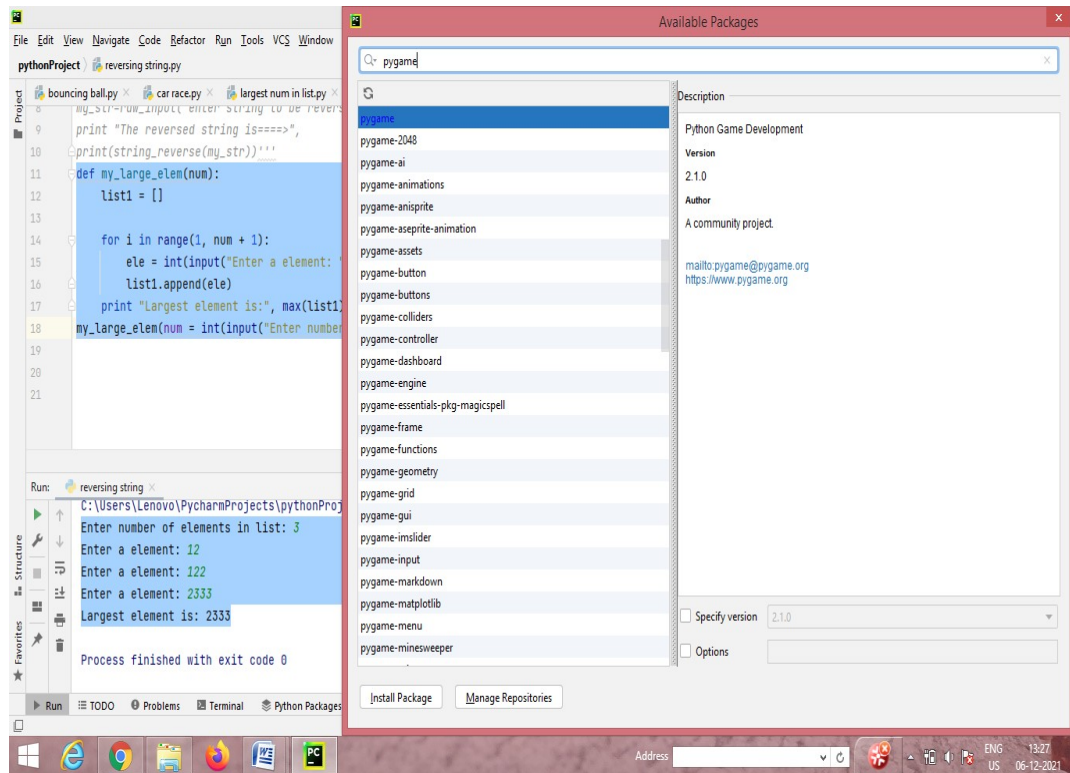
3. Click
pygame-1.9.3.tar.giz~2M and download Zar file
4. Extract the zar ile into C:\Python36-32\scripts older
5. Open command prompt
6. Type the following command
C:\>py -m pip install pygame—user
Collecting pygame
Downloading pygamee-1.9.3-cp36-cp36m-win32.whl(4.0MB)
After completion
Installing collected packages:pygame
Successfully installed pygame- 1.9.3

7. Now, pygame installed successfully
8. To see if it works, run one of the included examples in pygame-1.9.3
 - Open command prompt
 - Type the following

```
C:\>cd Python36-32\Scripts\pygame-1.9.3
C:\Python36-32\Scripts\pygame-1.9.3> cd
examples
C:\Python36-32\Scripts\pygame-
1.9.3\examples>aliens.py C:\Python36-
32\Scripts\pygame-1.9.3\examples>
```
9. Or
10. Goto file->settings->Click +



Type pygame in search bar and click install package at the bottom of the window



RESULT:

Thus the pygame tool is implemented successfully.

Ex.No: 12a	BOUNCING BALL
Date :	

AIM:

To write a python program to implement bouncing balls using pygame tool.

ALGORITHM:

1. Start the program.
2. Import the pygame package.
3. Initialize the pygame.
4. Display mode of pygame's window gets setted by assigning the values.
5. Within the game_loop(), block size and velocity are given. Position of x and y values are calculated by **pos_x=window_w/2, pos_y=window_h/2**.
6. Pygame updtes by calculating its velocity.
7. Stop the program.

PROGRAM:

```
import pygame
pygame.init()
window_w=800
window_h=600
white=(255,255,255)
black=(0,0,0)
FPS=120
window=pygame.display.set_mode((window_w,window_h))
pygame.display.set_caption("Game")
clock=pygame.time.Clock()
def game_loop():
    block_size=20
    velocity=[1,1]
    pos_x=window_w/2
    pos_y=window_h/2
    running=True
    while running:
        for event in pygame.event.get():
            if event.type==pygame.QUIT:
                pygame.quit()
                quit()
        pos_x+=velocity[0]
        pos_y+=velocity[1]
        if pos_x+block_size>window_w or pos_x<0:
            velocity[0]=-velocity[0]
        if pos_y + block_size > window_h or pos_y < 0:
            velocity[1] = -velocity[1]
        window.fill(white)
        pygame.draw.rect(window,black,[pos_x,pos_y,block_size,block_size])
        pygame.display.update()
        clock.tick(FPS)
game_loop()
```

OUTPUT:



RESULT:

Thus the python program to implement bouncing balls using pygame tool.

Ex.No: 12b	MOVING AN OBJECT ON KEYPRESS
Date :	

AIM:

To write a python program to implement moving an object on keypress using pygame tool.

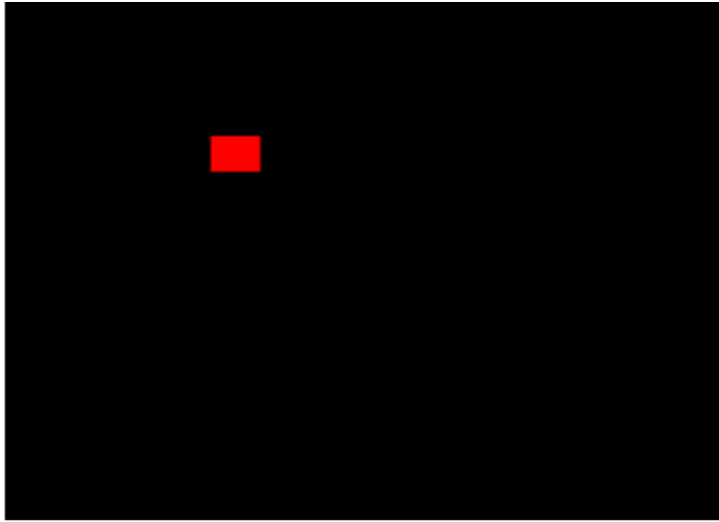
ALGORITHM:

1. Start the program.
2. Import the sys and pygame packages.
3. Set coordinates and size of the moving object.
4. Get the keys pressed.
5. If the left arrow key is pressed, it decrements in x axis.
6. If the right arrow key is pressed, it increments in x axis.
7. If the up arrow key is pressed, it decrements in y axis.
8. If the down arrow key is pressed, it increments in y axis.
9. Set screen background and place moving object on screen.
10. Displays the movement.
11. Stop the program.

PROGRAM:

```
import sys, pygame
pygame.init()
screen = pygame.display.set_mode((500, 500))
pygame.display.set_caption("Moving object")
# Moving object starting co-ordinates and size
x = 200
y = 200
width = 20
height = 20
# Moving speed
step = 5
# infinite loop
while 1:
    pygame.time.delay(10)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
    # Get the keys pressed
    keys = pygame.key.get_pressed()
    # Left arrow key - decrement in x axis
    if keys[pygame.K_LEFT] and x > 0:
        x -= step
    # Right arrow key - increment in x axis
    if keys[pygame.K_RIGHT] and x < 500 - width:
        x += step
    # Up arrow key - decrement in y axis
    if keys[pygame.K_UP] and y > 0:
        y -= step
    # Down arrow key - increment in y axis
    if keys[pygame.K_DOWN] and y < 500 - height:
        y += step
    # Set screen background and Place moving object on screen
    screen.fill((255, 255, 255))
    pygame.draw.rect(screen, (255, 0, 0), (x, y, width, height))
    # Display movement
    pygame.display.update()
```

OUTPUT:



RESULT:

Thus the python program to implement moving an object on keypress using pygame tool is executed.

Program for practice

Program 1

```
fn= input("enter source file name:")
data=file(fn).read()
words=data.split()
words.sort()
for word in words:
    print word
```

Output

```
enter source file name: 'inputfile.txt'
apple
control
doggy
hai
hello
python
sort
zebra
```

Program 2

```
def fact(n):
    if n==1:
        return n
    else:
        return n*fact(n-1)
n=int(input("enter a number"))
print("the factorial of", n,"is", fact(n))
```

Output

```
enter a number 5
('the factorial of', 5, 'is', 120)
```

Program 3

```
n=20
for i in range(2,n+1):
    k=0
    for j in range(2,i//2+1):
        if(i%j==0):
            k=k+1
    if(k<=0):
        print(i)
```

Output

```
2
3
5
7
11
13
17
19
```

Program 4

```
import time
ltime=time.localtime()
print(time.strftime("%a %b %d %H:%M:%S %Z %Y",ltime))
```

Output

Fri Feb 03 11:40:17 India Standard Time 2023

Program 5

```
c=int(input("enter temperature in celsius"))
f=((c*9/5)+32)
print("the equivalent temperature in fahernite+",f)
```

Output

enter temperature in celsius 32
('the equivalent temperature in fahernite+', 89)

Program 6

```
n=5
for i in range(1,n+1):
    print('*'*i)
for i in range(n,0,-1):
    print('*'*i)
```

Output

```
*
**
***
****
*****
*****
****
***
**
*
```

Program 7

```
str1="hello"
str2="world"
cstr=str1+str2
print("string1:",str1);
print("string2:",str2);
print("concatenate of strings:",cstr)
str3=cstr[3:5]
print("substring:",cstr[3:5])
```

Output

('string1:', 'hello')
('string2:', 'world')
('concatenate of strings:', 'helloworld')
('substring:', 'lo')

Program 8

```
n1=int(input("enter number1:"))
n2=int(input("enter number2:"))
print('' 1.addition
```

```

        2.subtraction
        3.multiplication
        4.division
        5.modulus'''
ch=int(input("enter your choice"));
if(ch==1):
    print(n1,'+',n2,'=',n1+n2)
elif(ch==2):
    print(n1,'-',n2,'=',n1-n2)
elif(ch==3):
    print(n1,'*',n2,'=',n1*n2)
elif(ch==4):
    print(n1,'/',n2,'=',n1/n2)
elif(ch==5):
    print(n1,'%',n2,'=',n1%n2)
else:
    print("error");

```

Output

```

enter number1:3
enter number2:7
1.addition
2.subtraction
3.multiplication
4.division
5.modulus
enter your choice2
(3, '-', 7, '=', -4)

```

Program 9

```

print(num3, 'is of type', type(num3))
num1=5
print(num1, 'is of type', type(num1));
num2=2.0
print(num2, 'is of type', type(num2));
num3=1+2j

```

Output

```

(5, 'is of type', <type 'int'>)
(2.0, 'is of type', <type 'float'>)
((1+2j), 'is of type', <type 'complex'>)

```