Ex.No:1a ELECTRICITY BILL CALCULATION

AIM

To Develop a flow chart for electricity billing by identifying and solving real life problem.

PROCEDURE

Step 1: Read the number of units consumed as units.

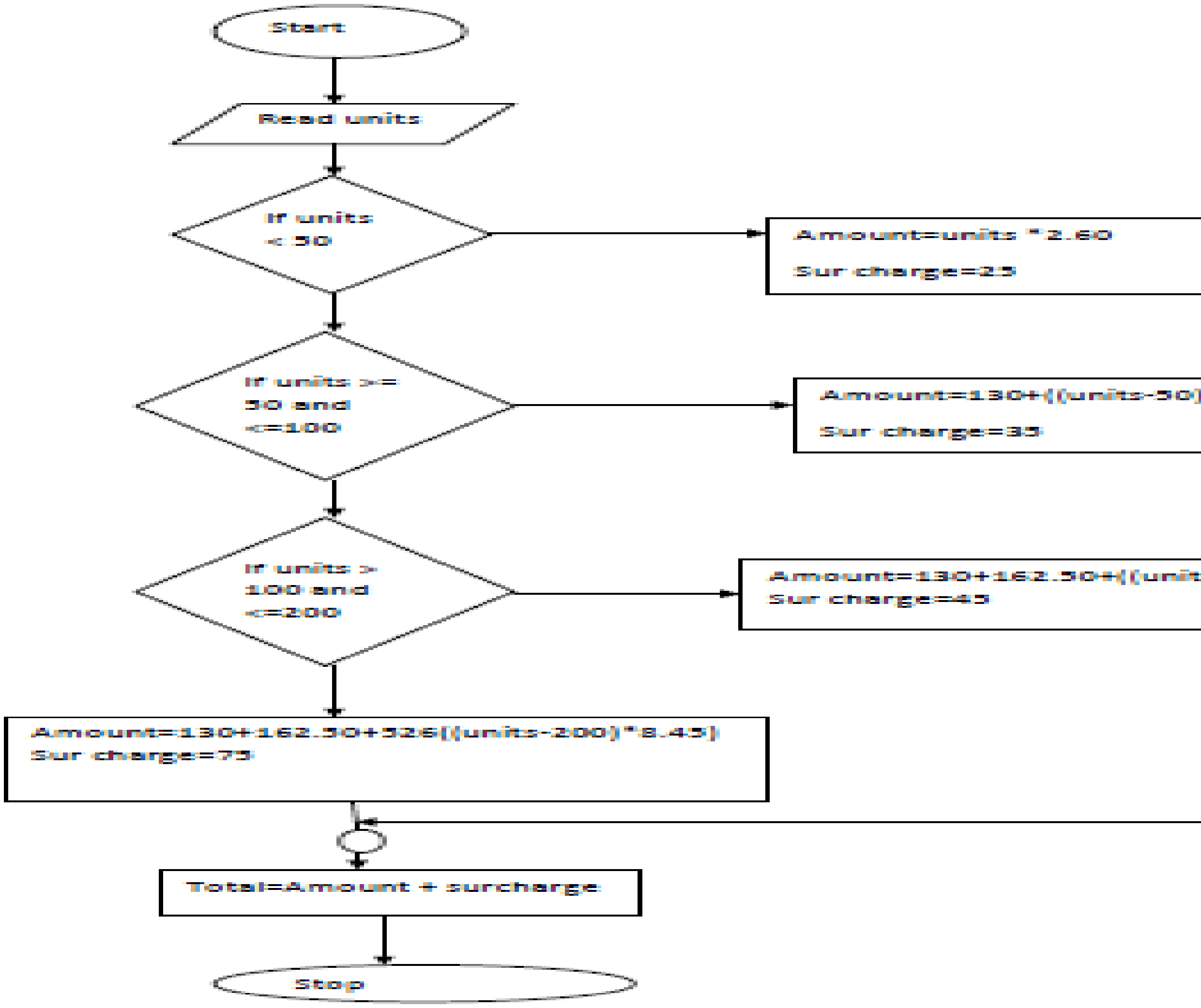
Step 2: Total amount (Rupees) = Amount + Surcharge

Step 3: Calculate the electricity bill as:

From Unit	To Unit	Amount	Surcharge	Maximum Unit
1	49	units *2.60	25	49
50	100	130+((units-50)*3.25)	35	100
101	200	130+162.50+((units-100)*5.26)	45	200
201	>500	130+162.50+526((units-200)*8.45)	75	>500

Step 4: Print the total amount to be paid for electricity bill.

FLOW CHART



RESULT

Thus the flowchart was drawn successfully.

Ex.No:1b RETAIL SHOP BILLING

AIM

To Develop a flow chart for Retail shop billing

PROCEDURE

Step 1: Calculate the tax.

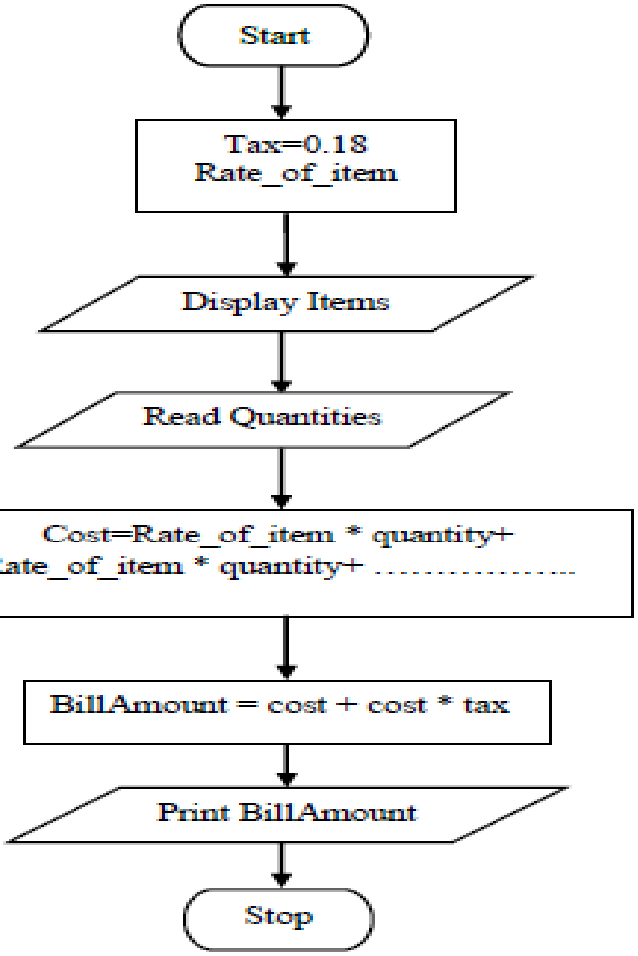
Step 2:Read the quantity of the product.

Step 3:Calculate the cost of the product.

Step 4: Calculate the total cost

Step 5: Print the total amount.

FLOW CHART


Start
Tax=0.18
Rate_of_item
Display Items
Read Quantities
Cost=Rate_of_item * quantity+
Rate_of_item * quantity+ ....
BillAmount = cost + cost * tax
Print BillAmount
Stop

RESULT

Thus the flowchart was drawn successfully.

Ex.No:1c SINE SERIES

AIM

To Develop a flow chart to calculate the sine series

PROCEDURE

Step 1: Read values for x and n.

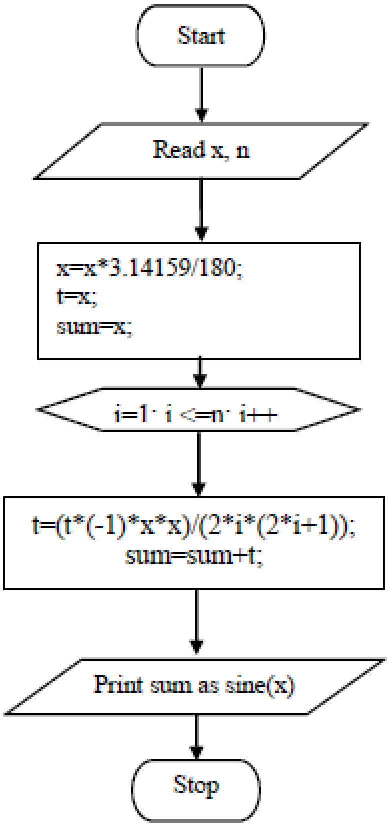
Step 2:Calculate x=x*3.14/180, t=x,sum=x

Step 3:Initialize the for loop.

Step 4: Calculate t=(t*(-1)*x*x)/(2*i*(2*i+1)) and sum=sum+t

Step 5: Print the result

FLOW CHART



RESULT

Thus the flowchart was drawn successfully.

Ex.No:1d WEIGHT OF A STEEL BAR

AIM

To Develop a flow chart to calculate the weight of a steel bar.

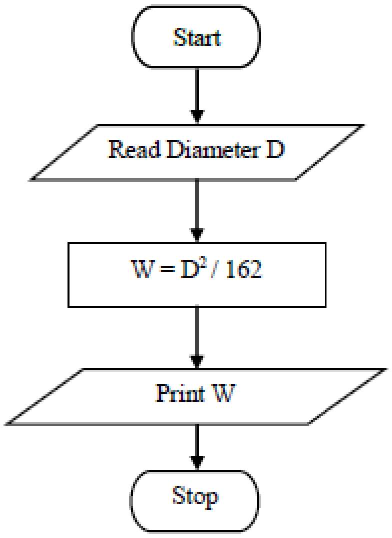
PROCEDURE

Step 1: Read the value for diameter.

Step 2:Calculate w=(d*d)/162

Step 3: Print the result

FLOW CHART



RESULT

Thus the flowchart was drawn successfully.

Ex.No:1e COMPUTE ELECTRICAL CURRENT IN A THREE PHASE CIRCUIT

AIM

To Develop a flow chart to calculate electrical current in a three phase circuit.

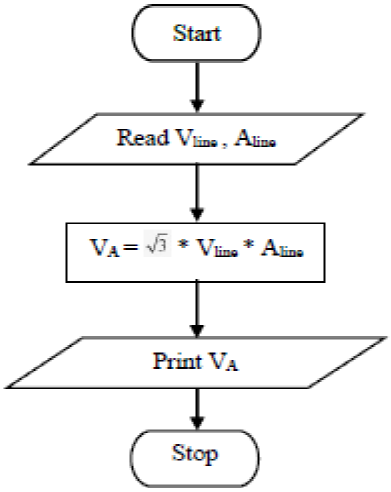
PROCEDURE

Step 1: Read the value for diameter.

Step 2:Calculate w=(d*d)/162

Step 3: Print the result

FLOW CHART


Start
Read V_{line} , A_{line}
$V_A = \sqrt{3} * V_{line} * A_{line}$
Print V_A
Stop

RESULT

Thus the flowchart was drawn successfully.

EX.NO:2a EXCHANGE THE VALUES OF TWO VARIABLES

AIM

To write the python program to exchange two variables

ALGORITHM

```
Step 1:  Start the program.
Step 2: Read the values for X and Y.
Step 3: Exchange the values of variables.
Step 4: Print the Exchanged values.
Step 5: Stop the Program.
```

PROGRAM

```
x=int(input("enter x:"))
y=int(input("enter y:"))
x,y=y,x
print("The value after swapping")
print("x=",x)
print("y=",y)
```

OUTPUT

```
enter x:2
enter y:6
The value after swapping
x= 6
y= 2
```

RESULT

Thus the program was executed successfully.

EX NO:2(b)(i) DISTANCE BETWEEN TWO POINTS

AIM

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

ALGORITHM

Step 1: Start the program.

Step 2: Read the values for X1,X2,Y1 and Y2.

Step 3: Calculate d=(((x2-x1)**2)+((y2-y1)**2))**0.5

Step 4: Print the Value for d

Step 5: Stop the Program.

PROGRAM

```
x1=int(input("enter x1 value:"))

x2=int(input("enter x2 value:"))

y1=int(input("enter y1 value:"))

y2=int(input("enter y2 value:"))

d=(((x2-x1)**2)+((y2-y1)**2))**0.5

print("The distance between two points :",d)
```

OUTPUT

```
enter x1 value:10

enter x2 value:12

enter y1 value:12

enter y2 value:14

The distance between two points : 2.8284271247461903
```

RESULT

Thus the program was executed successfully.

EX NO:2(b)(ii) DISTANCE BETWEEN TWO POINTS

AIM

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

ALGORITHM

Step 1: Start the program.

Step 2:Import math module

Step 3: Read the values for X1,X2,Y1 and Y2.

Step 3: Calculate d = math.sqrt(((p1[0]-p2[0])**2)+((p1[1]-p2[1])**2))
Step 4: Print the Value for d

Step 5: Stop the Program.

PROGRAM

```
import math

p1 = [4, 0]

p2 = [6, 6]

d = math.sqrt( ((p1[0]-p2[0])**2)+((p1[1]-p2[1])**2) )

print(d)
```

OUTPUT

```
6.324555320336759
```

RESULT

Thus the program was executed successfully.

EX NO 2(c) CIRCULATE THE VALUE OF N VARIABLES

AIM

To write a python program to circulate the value of N variables.

ALGORITHM

Step 1: Start the program.

Step 2: Define a function named rotate(l,n)

Step 3:Read the list of values.

Step 4: Call the rotate(l,n) function to circulate the values in clockwise and anti clockwise position.

Step 5: Print the circulated list

Step 6: Stop the program.

PROGRAM

```
def rotate(l,n):

   new_list=l[n:]+l[:n]

   return new_list

e1=[4,5,2,3,6,7,8]

print("original list=",e1)

m1=rotate(e1,1)

print("list rotated clockwise by 1",m1)

m1=rotate(e1,2)

print("list rotated clockwise by 2",m1)

m1=rotate(e1,-2)

print("list rotated anticlockwise by 2",m1)
```

OUTPUT

```
original list= [4, 5, 2, 3, 6, 7, 8]

list rotated clockwise by 1 [5, 2, 3, 6, 7, 8, 4]

list rotated clockwise by 2 [2, 3, 6, 7, 8, 4, 5]

list rotated anticlockwise by 2 [7, 8, 4, 5, 2, 3, 6]
```

RESULT

Thus the program was executed successfully.

EX:NO:3a NUMBER SERIES

AIM

To write the python program to read a number n and print and compute the series "1+2+...n=".

ALGORITHM

STEP 1: Start the program.

STEP 2: Take a value from the user and store it in a variable n.

STEP 3: Print the value of i and '+' operator while appending the value of i to a list.

STEP 4: Find the sum of elements in the list.

STEP 5: Print '=' followed by the total sum.

STEP 6: Stop the program.

PROGRAM

```
n=int(input("Enter a number: "))

a=[]

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

    print(i,end=" ")

    if(i<n):

        print("+”,end=" ")

    a.append(i)

print("=",sum(a))

print()
```

OUTPUT

```
Enter a number: 5

1+2+3+4+5=15
```

RESULT

Thus the program was executed successfully.

EX:NO:3b NUMBER PATTERNS

AIM

To write the python program to print the number patterns.

ALGORITHM

STEP 1: Start the program.

STEP 2: Print the pattern from the end of each Row

STEP 3: Complete the last column of each Row

STEP 4:Start from the Second Last Column of the second row

STEP 5:Repeat till the number of rows specified by the User.

STEP 6:Decide the number of rows and columns.
STEP 7: Stop the program.

PROGRAM

```
rows = int(input('Enter the number of rows:'))

for i in range(rows):

    for j in range(i):

        print(i, end=' ')

    print(")
```

OUTPUT

```
Enter the number of rows:8

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

6 6 6 6 6 6

7 7 7 7 7 7 7
```

RESULT

Thus the program was executed successfully.

EX:NO:3c PYRAMID PATTERNS

AIM

To write the python program to print the Pyramid pattern.

ALGORITHM

STEP 1: Start the program.

STEP 2: Accept the number of rows from the user to form pyramid shape.

STEP 3: Iterate the loop till the number of rows specified by the user.

STEP 4: Display 1 star in the first row.

STEP 5: Increase the number of stars based on the number of rows.

STEP 6: Stop the program.

PROGRAM

```
rows = int(input("Enter Pyramid Pattern Rows = "))

print("Pyramid Star Pattern")

for i in range(0, rows):

    for j in range(0, rows - i - 1):

        print(end = ' ')

    for k in range(0, i + 1):

        print('*', end = ' ')

    print()
```

OUTPUT

```
Enter Pyramid Pattern Rows = 7

Pyramid Star Pattern

      *

     * *

    * * *

   * * * *

  * * * * *

 * * * * * *

* * * * * * *
```

RESULT

Thus the program was executed successfully.

EX:NO:4 a OPERATIONS OF LIST

AIM:

To write the python program to create, slice, change, delete and index elements using lists .

ALGORITHM

STEP 1: Start the program.

STEP 2: Create the list.

STEP 3: Indexing the list using the index operator[].

STEP 4: Slicing the lists by using slicing operator – colon":".

STEP 5: Changing an element from the list.

STEP 6: Appending the lists.

STEP 7: Removing an element from the lists

STEP 8: Deleting an element from the list

STEP 9: Stop the program.

PROGRAM

```
my_list=['cement','bricks','sand','steel bar']

print(my_list[0])

print(my_list[2])

print(my_list[-3])

print(my_list[1:3])

print(my_list[0:])

my_list[1]= 'concrete'

print(my_list)

my_list.append('paint')
```

```
print(my_list)
del my_list[2]
print(my_list)
my_list.remove('steel bar')
print(my_list)
my_list[2:2]=['wiring','pipes']
print(my_list)
my_list.pop(1)
print(my_list)
print(my_list*3)
j=["Water","Wood"]
print(my_list+j)
```

OUTPUT

```
cement

sand

bricks

['bricks', 'sand']

['cement', 'bricks', 'sand', 'steel bar']

['cement', 'concrete', 'sand', 'steel bar']

['cement', 'concrete', 'sand', 'steel bar', 'paint']

['cement', 'concrete', 'steel bar', 'paint']

['cement', 'concrete', 'paint']

['cement', 'concrete', 'wiring', 'pipes', 'paint']

['cement', 'wiring', 'pipes', 'paint']

['cement', 'wiring', 'pipes', 'paint', 'cement', 'wiring', 'pipes', 'paint', 'cement', 'wiring', 'pipes',
'paint']

['cement', 'wiring', 'pipes', 'paint', 'Water', 'Wood']
```

RESULT

Thus the program was executed successfully.

EX:NO:4 b OPERATIONS OF TUPLES

AIM

To write the python program to create, slice, change, delete and index elements using Tuple.

ALGORITHM

STEP 1: Start the program

STEP 2: Create the tuple

STEP 3: Indexing the tuple using the index operator[].

STEP 4: Slicing the tuple by using the slicing operator – colon “:”.

STEP 5: Changing and iterating the tuple.

STEP 6: Count the tuple using count method.

STEP 7: Stop the program.

PROGRAM

```
my_tuple=("engine","transmission","battery","steering",

        "Brakes","FuelTank")

print(my_tuple)

print(my_tuple[0])

print(my_tuple[-1])

print(my_tuple[1:4])

#tuple membership test

print('engine' in my_tuple)

#iterate through a tuple

for components in (my_tuple):
    print("car component:",components)
print(my_tuple.index("engine"))
print(my_tuple.count("Brakes"))
```

OUTPUT

```
('engine', 'transmission', 'battery', 'steering', 'Brakes', 'FuelTank')
engine
FuelTank
('transmission', 'battery', 'steering')
True
car component: engine
car component: transmission
car component: battery
car component: steering
car component: Brakes
car component: FuelTank
0
1
```

RESULT

Thus the program was executed successfully.

EX: 5a IMPLEMENTING REAL-TIME/TECHNICAL APPLICATIONS USING SETS

AIM

To write a python program to implement the operation of sets (language).

ALGORITHM

```
Step1: start the program
Step2: Initialize the input as set1 and set2.
Step3:Print the set1 and set2.
Step4: Print the union of  set1 and set2 by doing operation “set1|set2” and store it in set3.
Step5: Print the intersection of set1 and set2 by doing operation “set1&set2” andstore it in set4.
Step4: check the relationship between set3 and set4
        (i) if set3 >set4 then print "Set3 is superset of Set4".
        (ii) elif set3<set4 then print "Set3 is subset of Set4".
        (iii) else set3=set4 then print "Set3 is same as Set4".
Step 5: Print the difference between set1 and set2by doing operation “set3-set4” and store it in set5.
Step6: To remove all the values in set5 by function set5.clear()
Step 7: Stop the program
```

PROGRAM

```
set1 = {"Python", "Java", "C"}
set2 = {"Java", "Php", "Python"}
print("Set1 = ", set1)
print("Set2 = ", set2)
print("\n")
set3 = set1 | set2
print("Union of Set1 & Set2: Set3 = ", set3)
set4 = set1 & set2
print("Intersection of Set1 & Set2: Set4 = ", set4)
print("\n")
if set3 > set4:
    print("Set3 is superset of Set4")
elif set3 < set4:
    print("Set3 is subset of Set4")
else :
    print("Set3 is same as Set4")
set5 = set1 - set2
print("Elements in Set3 and not in Set4: Set5 = ", set5)
print("\n")
set5.clear()
print("After applying clear on sets Set5: ")
print("Set5 = ", set5)
```

OUTPUT

```
Set1 =  {'Java', 'Python', 'C'}
Set2 =  {'Java', 'Python', 'Php'}
Union of Set1 & Set2: Set3 =  {'Java', 'Php', 'Python', 'C'}
Intersection of Set1 & Set2: Set4 =  {'Java', 'Python'}
Set3 is superset of Set4
Elements in Set3 and not in Set4: Set5 =  {'C'}
After applying clear on sets Set5:
Set5 =  set()
```

RESULT

Thus the program was executed successfully.

EX: 5b IMPLEMENTING REAL-TIME/TECHNICAL APPLICATIONS USING DICTIONARY

AIM

To write a python program to implement the operation of dictionary (automobile details).

ALGORITHM

STEP 1: Start the program

STEP 2: Create the dictionary using my_dict[].

STEP 3: Change an element to dictionary.

STEP 4: Remove an element to dictionary using .pop() and length of the key values using len().

STEP 5: Delete an element to dictionary.

STEP 7: Stop the program.

PROGRAM

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = car.copy()
print(x)
print("\n")
x = car.get("model")
print("model:",x)
print("\n")
x = car.items()
print(x)
print("\n")
x = car.keys()
print(x)
```

```
car["color"] = "white"
print(x)
print("\n")
car.popitem()
print(car)
print("\n")
car.update({"color": "Red"})
print(car)
print("\n")
x = car.values()
print(x)
print("\n")
print("length of car dictionary=",len(car))
print("\n")
car.clear()
print(car)
```

OUTPUT

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
model: Mustang
dict_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)])
dict_keys(['brand', 'model', 'year'])
dict_keys(['brand', 'model', 'year', 'color'])
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'Red'}
dict_values(['Ford', 'Mustang', 1964, 'Red'])
length of car dictionary= 4
```

RESULT

Thus the program was executed successfully.

EX: 6a IMPLEMENTING PROGRAM USING FUNCTION(FACTORIAL)

AIM

To write a python program to find the factorial of a number.

ALGORITHM

```
Step 1: Start the program
Step 2:Define a factorial function.
Step 3: Initialize Variable fact=1.
Step 4: Using for loop for finding factorial
Step 5: Return factorial.
Step 6: Read the input from the user.
Step 7: Function call and assign the value to variable result
Step 8: Print the factorial of number,
Step 9: Stop
```

PROGRAM

```
def factorial(num):
    fact=1
    for i in range(1, num+1):
        fact=fact*i
    return fact
number=int(input("Please enter any number to find factorial: "))
result=factorial(number)
print("The factorial of",number,"is" ,result)
```

OUTPUT

```
Please enter any number to find factorial: 5
The factorial of 5 is 120
```

RESULT

Thus the program was executed successfully.

EX :6b IMPLEMENTING PROGRAM USING FUNCTION(LARGEST NUMBER IN A LIST)

AIM

To write a python program to find the largest number in a list.

ALGORITHM

Step 1: Start the program

Step 2:Define a myMax(list1).
Step 3: Assume first number in list is largest initially and assign it to variable "max".

Step 4: Now traverse through the list and compare each number with "max" value, whichever is largest assign that value to "max'.

Step 5: after complete traversing the list return the "max" value

Step 6: define the number in a list.
Step 7: Print the largest number in a list.
Step 8: Stop the program.

PROGRAM

```
def myMax():
   list1=[]
   n=int(input("enter the number of elements in list"))
   for i in range(n):
       a=int(input("enter the values:"))
       list1.append(a)
   list1.sort()
   max=list1[n-1]
   return(max)
print("Largest element is:", myMax())
```

OUTPUT

```
enter the number of elements in list5
enter the values:10
enter the values:52
enter the values:4
enter the values:10
enter the values:45
Largest element is: 52
```

RESULT

Thus the program was executed successfully.

EX :6c IMPLEMENTING PROGRAM USING FUNCTION(AREA OF SHAPE)

AIM:

To write a python program to find the largest number in a list.

ALGORITHM:

```
Step 1: Start the program
Step 2:Define a function areacalculator().
Step 3: Read the input from the user.
Step 4: Initialize area=0 , pie=3.14 .
Step 5: if the input is "Square", get value of side from the user and calculate the area of square.
Step 6: else if the input is "Circle", get the values of radius from the user and calculate the area of circle.
Step 7: else if the input is "Rectangle", get the values of length and width from the user and calculate the area of rectangle.
Step 8: else if the input is "Triangle", get the values of base and height from the user and calculate the area of rectangle.
Step 9: else print select a valid input.
Step 10: stop the program.
```

PROGRAM

```
def areacalculator():
    shape = input("Enter the shape you want to calculate area of: ")
    area = 0
    pie = 3.14
    if shape== "Square":
        side = int(input("Enter the value of side: "))
        area = area + (side ** 2)
    elif shape== "Circle":
        radius = int(input("Enter the value of radius: "))
        area = area + (2 * pie * radius)
    elif shape== "Rectangle":
        length = int(input("Enter the value of length: "))
        width = int(input("Enter the value of length: "))
        area = area + (length * width)
    elif shape== "Triangle":
        base = int(input("Enter the value of base: "))
        height = int(input("Enter the value of height: "))
        area = area +(0.5 * base * height)
    else:
        print ("Select a valid shape")
    print ("%.2f" % area)
areacalculator()
```

OUTPUT

```
Enter the shape you want to calculate area of : Square

Enter the value of  side: 7

49.00
```

RESULT

Thus the program was executed successfully.

EX: 7a IMPLEMENTING PROGRAM USING STRINGS(REVERSE)

AIM

To write a python program to reverse the string.

ALGORITHM

```
Step1: start the program
Step2: Read the string
Step3:calculate length of the list
Step4: using slice syntax reverse a string
Step5: print the reversed string
Step 6: stop the program
```

PROGRAM

```
s="Python"
stringlength=len(s)
slicedString=s[stringlength::-1]
print (slicedString)
```

OUTPUT

```
nohtyP
```

RESULT

Thus the program was executed successfully.

EX :7b IMPLEMENTING PROGRAM USING STRINGS(PALINDROME)

AIM

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

ALGORITHM

Step 1: start the program

Step 2:Read the letter from a user.

Step 3:Compare the temporary variable with reverses letter.

Step 4:If both letters are the same, print "this string is a palindrome."

Step 5:Else print, "This string is not a palindrome."

Step 6: stop the program

PROGRAM

```
string=input(("Enter a letter:"))
if(string==string[::-1]):
   print("The letter is a palindrome")
else:
  print("The letter is not a palindrome")
```

OUTPUT

```
Enter a letter:PYTHON
The letter is not a palindrome
```

RESULT

Thus the program was executed successfully.

EX :7c IMPLEMENTING PROGRAM USING STRINGS(COUNT)

AIM

To write a python program to count the character of the string.

ALGORITHM

Step 1:start the program

Step 2:Define a string.

Step 3:Define and initialize a variable count to 0.

Step 4:Iterate through the string till the end and for each character except spaces, increment the count by 1.

Step 5:To avoid counting the spaces check the condition i.e. string[i] != ' '.

Step 6:Displays the total number of characters present in the given string

Step 7: stop the program

PROGRAM

```
string = "count the character";
count = 0;
for i in range(0, len(string)):
if(string[i] != ' '):
count = count + 1;
print("Total number of characters in a string: " + str(count));
```

OUTPUT

```
Total number of characters in a string: 17
```

RESULT

Thus the program was executed successfully.

EX: 7d IMPLEMENTING PROGRAM USING STRINGS(REPLACE)

AIM

To write a python program to replace the character of the string.

ALGORITHM

Step 1:Start the program

Step 2:Define a string.

Step 3:Define a new string using string replace method.

Step 4:Print the original string.

Step 5:Print the replaced string.

Step 6: Stop the program

PROGRAM

```
input_string = "python"
new_string = input_string.replace('p', "P")
print("The original string is:")
print(input_string)
print("The replaced String is:")
print(new_string)
```

OUTPUT

```
The original string is:
python
The replaced String is:
Python
```

RESULT

Thus the program was executed successfully.

Ex.No:8a PANDAS LIBRARY

AIM

To write a python program to implement the Pandas Library

ALGORITHM

Step1: start the program

Step2: Import the pandas library using the alias pd.

Step 3: Initialize the inputs in dataset k1.

Step 4: Pandas DataFrame will be created by loading the datasets.

Step 5: Two-dimensional data structure is aligned in a tabular fashion in rows and columns and stored in jk.

Step 6: Print the table for manipulating data.

PROGRAM

```
import pandas as pd

kl = {'Name':["Sam","peter","mary"],'Age':[10,20,30]}

jk = pd.DataFrame(kl)

print(jk)
```

OUTPUT

Open command prompt :

Type **pip install pandas** and click enter

Then run the module

```
       Name     age
 0      Sam      10
 1      Peter    20
 2      Mary     30
```

RESULT

Thus the program was executed successfully.

Ex.No:8b **Numpy Library**

AIM

To write a python program to implement the Numpy Library

ALGORITHM

```
Step1: start the program
Step2: Import the numpy library using the alias np.
Step 3: Create ndarray to store values using array()
Step 4: Complex type of objects assigned using type()
Step 5: Print the complex number
```

PROGRAM

```
import numpy as np

realtocom=np.array([1,2,3,4,5],dtype=complex)

print(realtocom)
```

OUTPUT

Open command prompt :

Type **pip install numpy** and click enter

Then run the module

```
 [1.+0.j 2.+0.j 3.+0.j 4.+0.j 5.+0.j]
```

RESULT

Thus the program was executed successfully.

Ex.No:8c **MATPLOTLIB LIBRARY**

AIM

To write a python program to plot a graph using matplotlib library.

ALGORITHM

STEP 1: Start the program.

STEP 2: Define the x-axis and corresponding y-axis values as lists.

STEP 3: Plot them on canvas using .plot() function

STEP 4: Give a name to x-axis and y-axis using .xlabel() and .ylabel() functions.

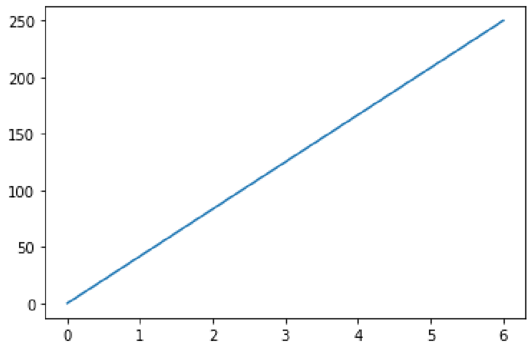
STEP 5: Finally, to view your plot, use .show() function.

STEP 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 program was executed successfully.

Ex.No:8d SCIPY LIBRARY

AIM

To write a python program to implement the Numpy Library

ALGORITHM

Step1: start the program
Step2: Import the module cosine from scipy.spatial.distance using scipy library.
Step 3: Read the values for two points p1 and p2.
Step 4: Find the cosine value for those points and stored in res.
Step 5: Print the resultant cosine value.

PROGRAM

```
from scipy.spatial.distance import cosine
p1 = (1, 0)
p2 = (10, 2)
res = cosine(p1, p2)
print(res)
```

OUTPUT

Open command prompt :
Type **pip install scipy** and click enter
Then run the module

```
0.019419324309079777
```

RESULT

Thus the program was executed successfully.

Ex.No:9a COPY FILE ONE FILE TO ANOTHER FILE

AIM

To write a python program Copy contents of one file to another file.

ALGORITHM

Step 1: Create source file with content and target file without content.
Step 2: Get the input file using input() function.
Step 3:File is opened using the open() function in the read mode.
Step 4: Read all lines in source file through readlines() and stored in a variable.
Step 5: Open the target file and use for loop to write all lines in the target file.
Step 6: File is copied from source file to target file.

PROGRAM

```
import time
print("Enter the Name of Source File: ")
sFile = input()
print("Enter the Name of Target File: ")
tFile = input()
fileHandle = open(sFile, "r")
texts = fileHandle.readlines()
fileHandle.close()
fileHandle = open(tFile, "w")
for s in texts:
fileHandle.write(s)
fileHandle.close()
print("File Copied Successfully!")
```

OUTPUT:

```
Enter the Name of Source File:
Copytext.txt
Enter the Name of Target File:
Textcopy.txt
File Copied Successfully
```

RESULT

Thus the program was executed successfully.

Ex.No:9b COUNTING NUMBER OF WORDS IN A FILE

AIM

To write a python program to counting number of words in a file using file handling.

ALGORITHM

Step 1: Enter a file name.
Step 2: The file is opened using the open() function in the read mode.
Step 3: A for loop is used to read through each line in the file.
Step 4: Each line is split into a list of words using split().
Step 5: Number of words in each line is counted using len() and the count variable is incremented.
Step 6: Number of words in the file is printed.

PROGRAM

```
import time

fname = input("Enter file name: ")

num_words = 0

with open(fname, 'r') as f:

for line in f:

words = line.split()

num_words += len(words)

print("Number of words:")

print(num_words)
```

OUTPUT:

```
Enter file name: file1.txt

Number of words:5
```

RESULT

Thus the program was executed successfully.

Ex.No:9c LONGEST WORD IN A FILE

AIM

To write a python program to find longest word in a file using file handling.

ALGORITHM

```
Step 1:Create a file with more number of words.
Step 2: File is opened using the open() function in the read mode.
Step 3: Read the file through read().
Step 4: Each line is split into a list of words using split() and stored in words.
Step 5: Number of words in each line is counted and find length using len(max(words,key=len)).
Step 6: Longest word in the file is printed.
```

PROGRAM

```
import time
deflongestword(filename):
        with open(filename,'r+') as f:
                words = f.read().split()
                max_len_word = max(words,key=len)
                max_len = len(max(words,key=len))
                print('maximum lenth word in file :',max_len_word)
                print('lenth is : ',max_len)
longestword('file1.txt')
```

OUTPUT:

```
Maximum length word in file : programming

Length is : 11
```

RESULT

Thus the program was executed successfully.

Ex.No:10a HANDLING DIVIDE BY ZERO EXCEPTION

AIM

To write a python program to handling divide by zero error.

ALGORITHM

```
Step 1:Take inputs from the user, two numbers.
Step 2: If the entered data is not integer, throw an exception.
Step 3: If the remainder is 0, throw divide by zero exception.
Step 4: If no exception is there, return the result.
```

PROGRAM

```
import time

try:

   num1 = int(input("Enter First Number: "))

   num2 = int(input("Enter Second Number: "))

result = num1 / num2

print(result)

exceptValueError as e:

print("Invalid Input Please Input Integer...")

exceptZeroDivisionError as e:

print(e)
```

OUTPUT

```
Enter First Number: 12

Enter Second Number: 0

division by zero
```

RESULT

Thus the program was executed successfully.

Ex.No:10b CHECK VOTER’S AGE ELIGIBILITY USING EXCEPTION HANDLING

AIM

To write a python program to Check voter’s age eligibility using exception handling.

ALGORITHM

Step 1: Get the age using input() function in try block.

Step 2: Validate the age through if condition. Condition is true print if-statement , otherwise print else- statement.

Step 3:Input is invalid automatically go to except block.

Step 4:finally function calling**main()**.

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("Exception handled")
        print("age must be a valid number")
main()
```

OUTPUT

```
Enter your age26

Eligible to vote
```

RESULT

Thus the program was executed successfully.

Ex.No:11 EXPLORING PYGAME TOOL

AIM

To write a python program to exploring the pygame tool

ALGORITHM

Step1: start the program

Step2: Import the pygame library to create video games by accessing all functions of pygame.

Step 3: Initialize all the required module of the pygame using pygame.init() function.

Step 4: Move the player by pressing the key up, down, left and right arrows.

Step 5: Hitting the player by collision using spritecollide() function.

Step 6: To track the time in pygame clock by clock.tick() function.

Step 7: Stop the pygame by pygame.quit().

PROGRAM

```
import pygame

import sys

#Sprite class

class Sprite(pygame.sprite.Sprite):

    def __init__(self, pos):

        pygame.sprite.Sprite.__init__(self)

        self.image = pygame.Surface([20, 20])

        self.image.fill((255, 0, 255))

        self.rect = self.image.get_rect()

        self.rect.center = pos
```

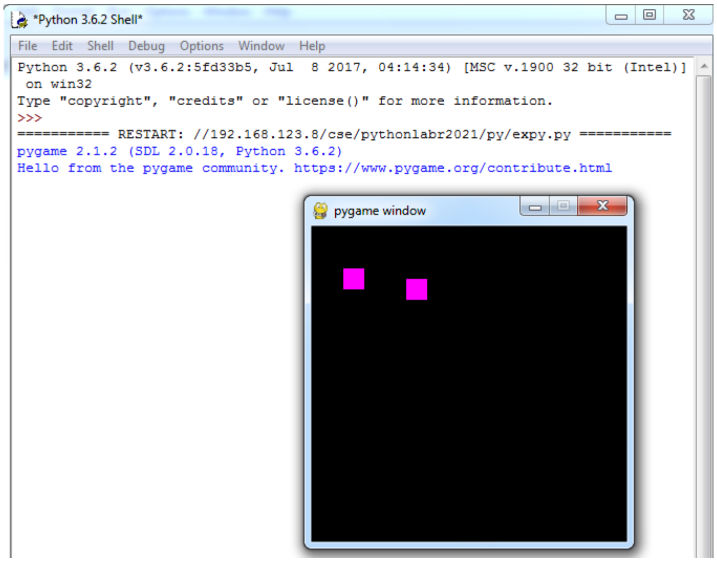
```
def main():

    pygame.init()

    clock = pygame.time.Clock()

    fps = 50

    bg = [0, 0, 0]

    size =[300, 300]

    screen = pygame.display.set_mode(size)

    player = Sprite([40, 50])

    # Define keys for player movement

    player.move = [pygame.K_LEFT, pygame.K_RIGHT, pygame.K_UP,
pygame.K_DOWN]

    player.vx = 5

    player.vy = 5

      wall = Sprite([100, 60])

      wall_group = pygame.sprite.Group()

    wall_group.add(wall)

      player_group = pygame.sprite.Group()

    player_group.add(player)

      while True:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                return False

        key = pygame.key.get_pressed()

        for i in range(2):
            if key[player.move[i]]:
                player.rect.x += player.vx * [-1, 1][i]
          for i in range(2):
            if key[player.move[2:4][i]]:
                player.ret.y += player.vy * [-1, 1][i]
        screen.fill(bg)
        # first parameter takes a single sprite
        # second parameter takes sprite groups
        # third parameter is a kill command if true
        hit = pygame.sprite.spritecollide(player, wall_group, True)
        if hit:
        # if collision is detected call a function to destroy
            # rect
            player.image.fill((255, 255, 255))
        player_group.draw(screen)
        wall_group.draw(screen)
        pygame.display.update()
        clock.tick(fps)
    pygame.quit()
    sys.exit
main()
```

OUTPUT

Open command prompt :

Type **pip install pygame** and click enter

Then run the module



```
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
 on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=========== RESTART: //192.168.123.8/cse/pythonlabr2021/py/expy.py ===========
pygame 2.1.2 (SDL 2.0.18, Python 3.6.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

RESULT

Thus the program was executed successfully.

EX NO 12 SIMULATE BOUNCING BALL USING PYGAME

AIM

To write a Python program to bouncing ball in Pygame.

ALGORITHM

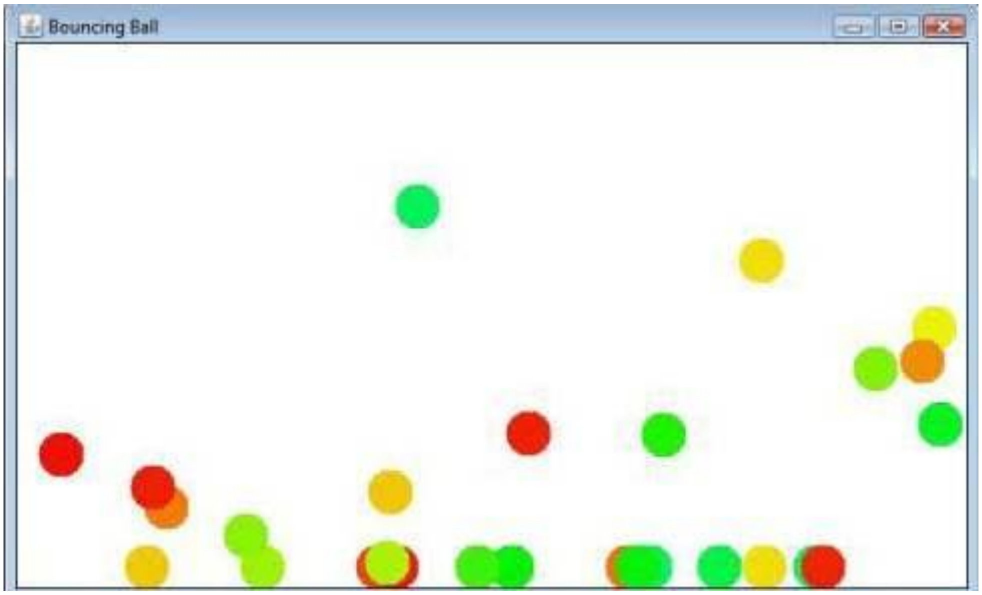
1. Import the required packages
2. Define the required variables
3. Define the screen space to display the bouncing balls in that space

PROGRAM

```
import sys

import pygame

pygame.init()

size = width, height = 320, 240

speed = [2, 2]

black = 0, 0, 0

screen = pygame.display.set_mode(size)

ball = pygame.image.load('C:\\Users\\admin\\Desktop//ball.jpg')

ballrect = ball.get_rect()

while 1:

   for event in pygame.event.get():

      if event.type == pygame.QUIT: sys.exit()


   ballrect = ballrect.move(speed)
```

```
if ballrect.left < 0 or ballrect.right > width:
   speed[0] = -speed[0]
if ballrect.top < 0 or ballrect.bottom > height:
   speed[1] = -speed[1]
screen.fill(black)
screen.blit(ball, ballrect)
pygame.display.flip()
```

OUTPUT



RESULT

Thus the program was executed successfully.