

Expt. No.....!

Page No.....!

## SIMPLE INTEREST

AIM:

To display the ~~values~~, ~~max~~ Simple interest

CODING:

```
p = float(input("Enter the principal amount "))  
n = float(input("Enter the no. of years "))  
r = float(input("Enter the Rate of interest "))  
s.i = (p * n * r) / 100  
print("The Simple Interest is ", s.i)
```

OUTPUT:

Enter the principal amount 2000

Enter the no. of years 2

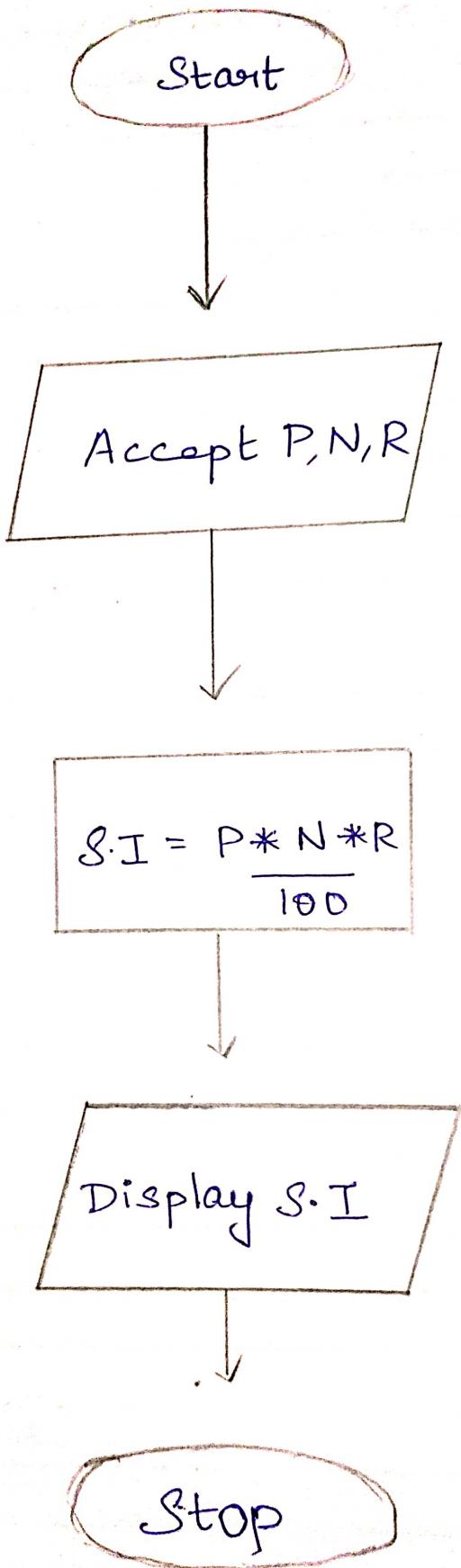
Enter the rate of interest 4

The Simple Interest is 160.0

✓

Project

## FLOWCHART:



Expt. No.....2.....

Page No.....2.....

### TIME

#### AIM:

To display the hours, mins and Seconds

#### CODING:

```
s = int(input("Enter the time in seconds"))
```

```
h = s / 3600
```

```
s = s % 3600
```

```
m = s / 60
```

```
s = s % 60
```

```
print("The hours, minutes, seconds is", h, m, s, sep=":")
```

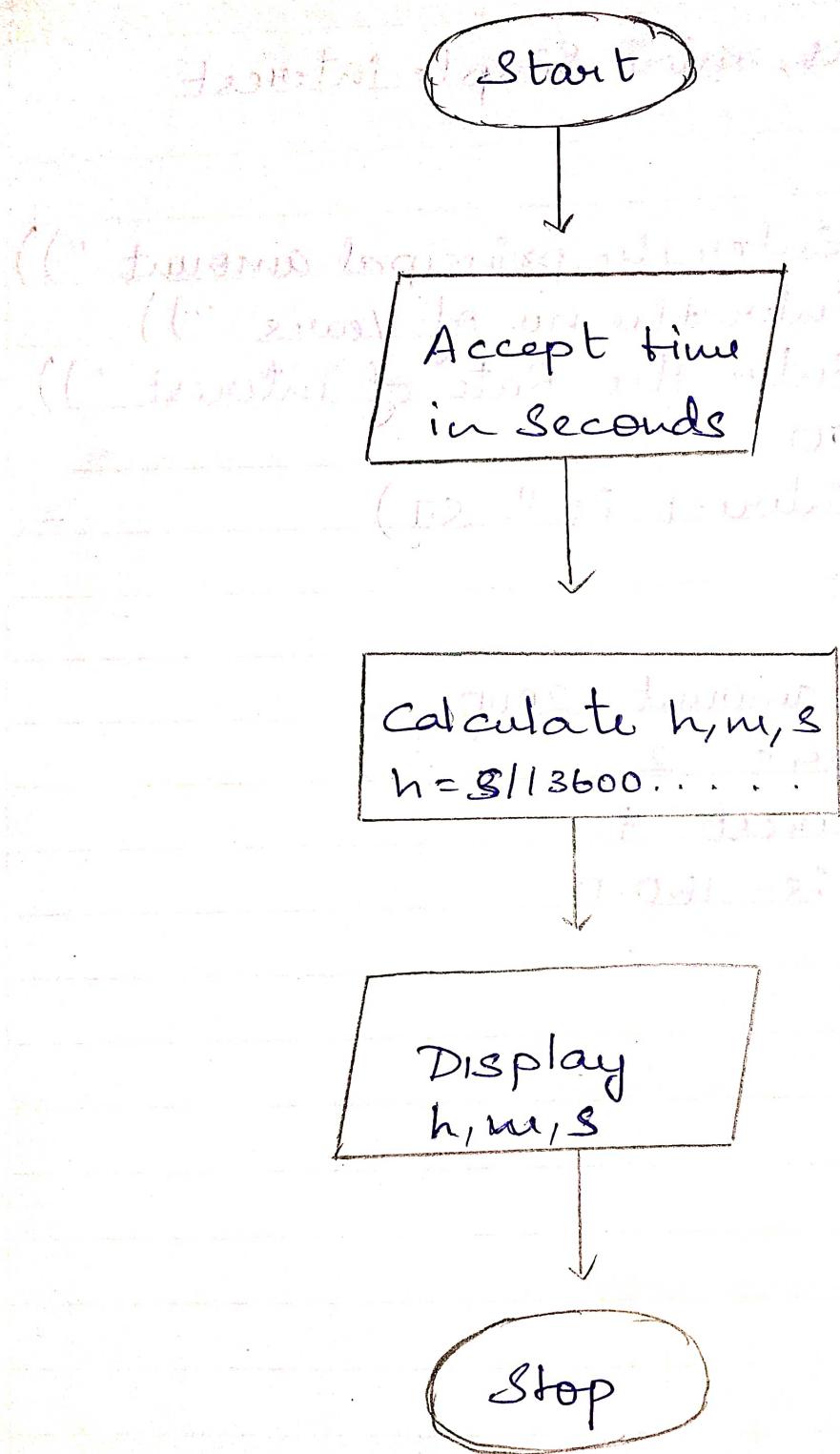
#### OUTPUT:

Enter the time in seconds 9000

The hours, minutes, seconds is 2: 30: 0

c  
D

## FLOWCHART:



AVERAGE MARK

AIM:

To display the average mark and Grade

CODING:

Solve:

Print("Enter the marks of 5 subjects")

Sub1 = float(input())

Sub2 = float(input())

Sub3 = float(input())

Sub4 = float(input())

Sub5 = float(input())

Tot\_mark = Sub1 + Sub2 + Sub3 + Sub4 + Sub5

avg = Tot\_mark / 5

print(avg)

if avg &gt; 90:

print("A")

elif avg &lt; 90 &amp; avg &gt;= 80:

print("B")

elif avg &lt; 80 &amp; avg &gt;= 70:

print("C")

elif avg &lt; 70 &amp; avg &gt;= 60:

print("D")

else:

print("You have Failed")

P.T.O

Expt. No.....

Page No.....

**OUTPUT:**

Enter the marks of 5 subject

95.5

92.5

99.5

96.5

96

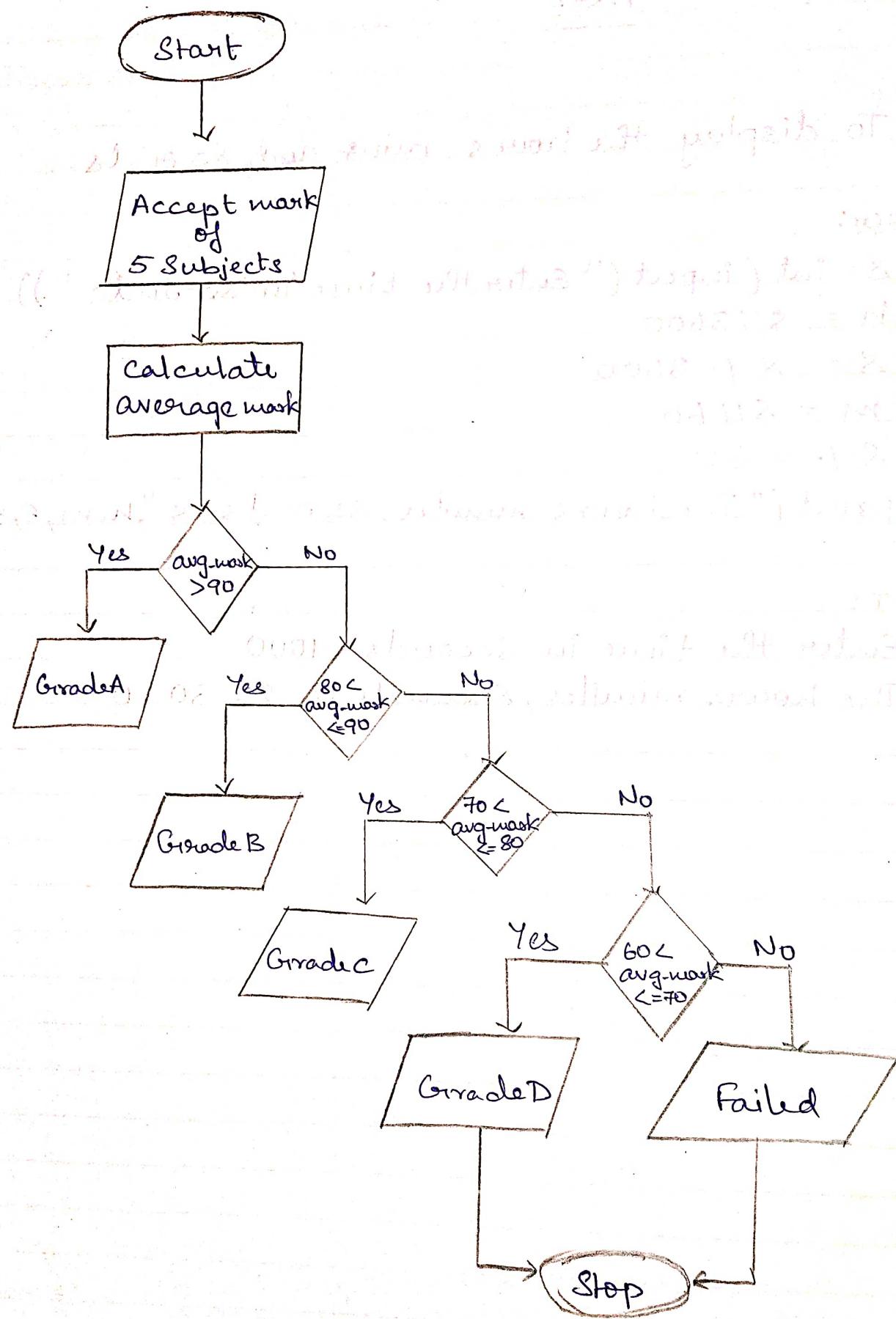
96 ✓

'A'

Q  
D ✓

90+%

## FLOWCHART:



Expt. No.....4.....

Page No.....9.....

## CELSIUS AND FARENHEIT

### AIM:

To display the celsius and farenheit

### CODING:

```
c=float(input("Enter the temperature in Celsius"))
a=((9/5)*c)+32
print("The temperature is",a,"Farenheit")
f=float(input("Enter the temperature in Farenheit"))
b=(f-32)*(5/9)
print("The temperature is",b,"degrees")
```

### OUTPUT:

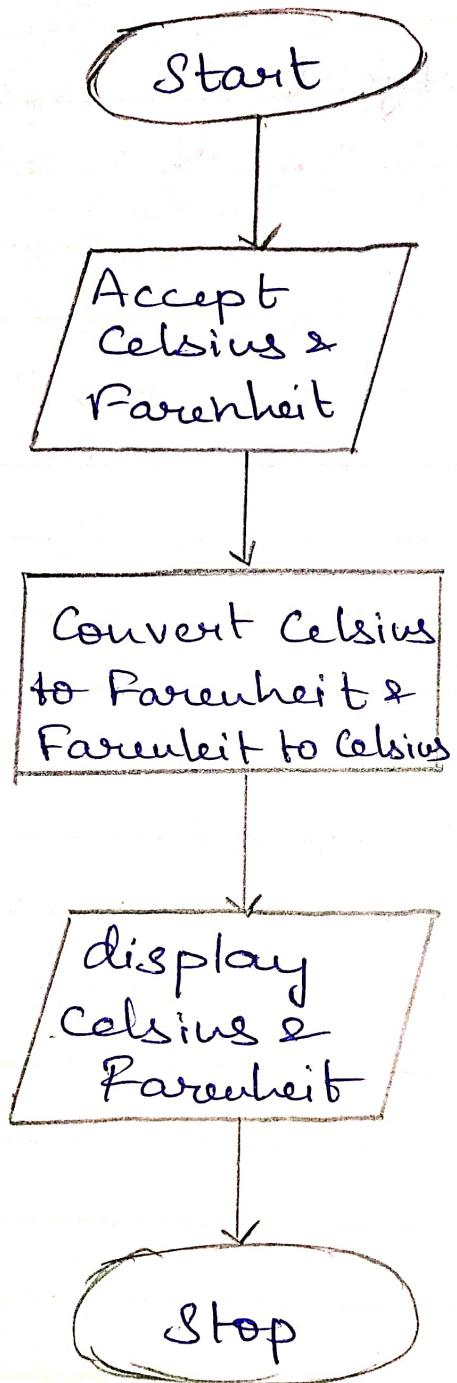
Enter the temperature in Celsius -40

The temperature is -40 Farenheit

Enter the temperature in Farenheit -40

The Temperature is -40 degrees.

## Flowchart:



## AREA AND PERIMETER OF CIRCLE

## AIM:

To display area or perimeter depending on the user's choice.

## CODING:

```
r = float(input("Enter the radius of the circle "))
print("1 - Area")
print("2 - perimeter")
print("3 - exit")
x = int(input())
if x == 1:
    ar = 3.14 * (r * r)
    print("Area = ", ar)
elif x == 2:
    perimeter = 2 * 3.14 * r
    print("Perimeter = ", perimeter)
elif x == 3:
    print("Exit")
```

## OUTPUT:

Enter the radius of the circle ?

| 1 - Area |

| 2 - perimeter |

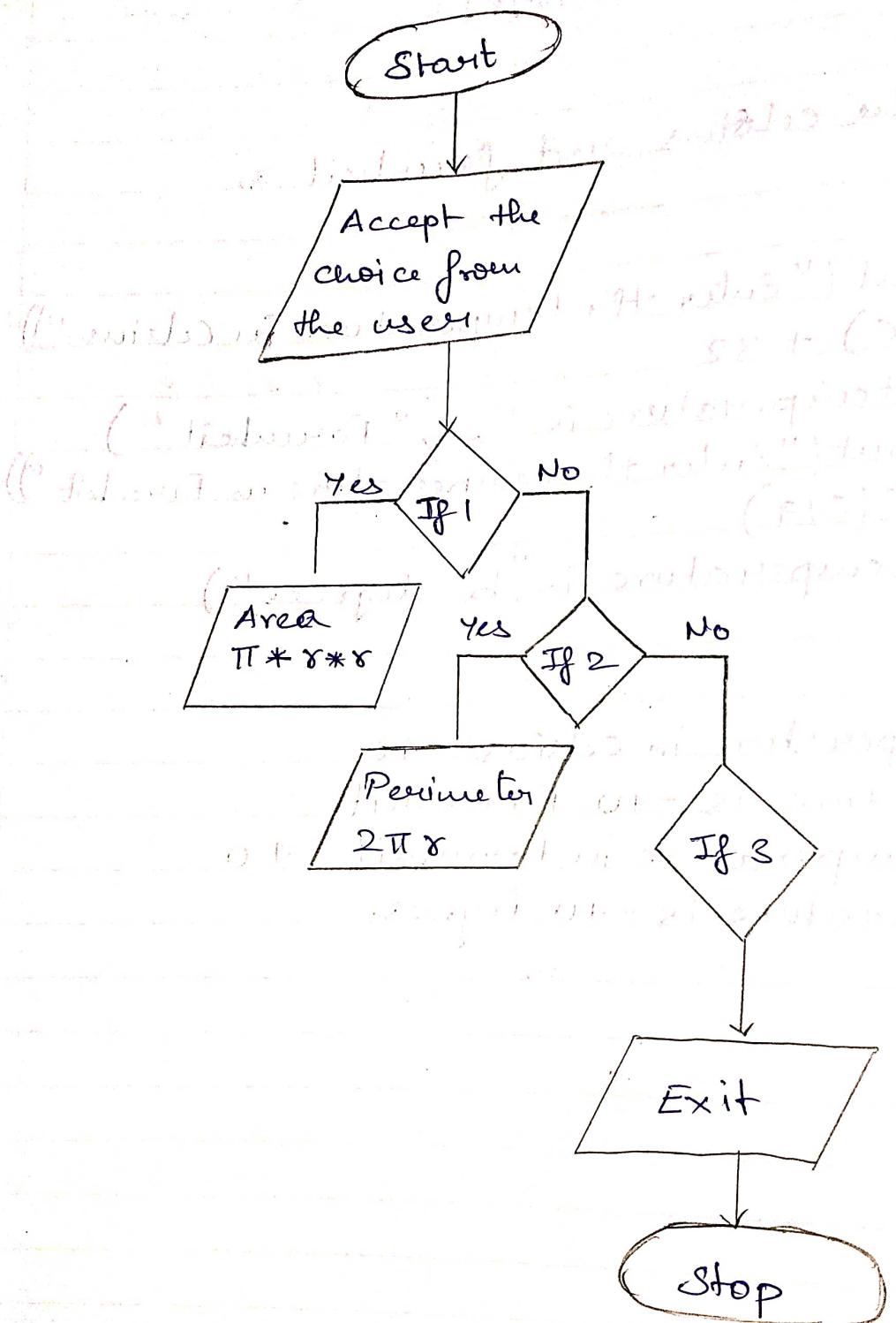
| 3 - exit |

perimeter 2

perimeter = 43.96

THAN

## FLOWCHART :



## LEAP YEAR

AIM:

To display whether the given year is a leap year or not.

CODING:

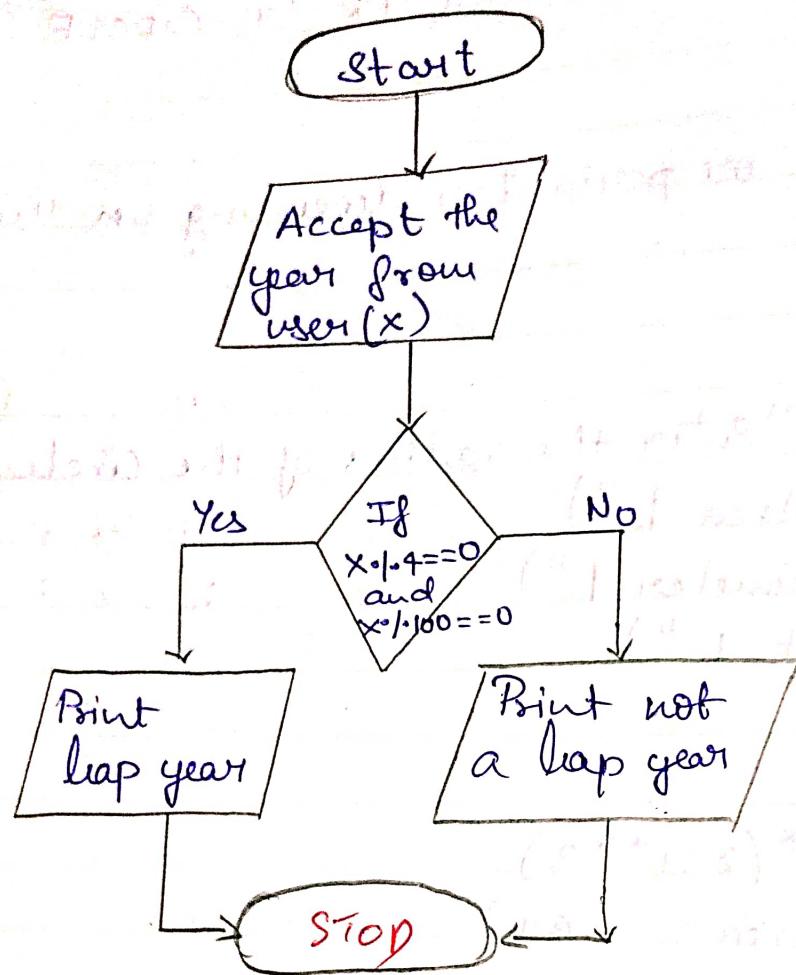
```
x = int(input("Enter a year"))
if x%4 == 0 and x%100 != 0 or x%400 == 0:
    print("It is a leap year")
else:
    print("not a leap year")
```

OUTPUT:

Enter a year 2020

It is a leap year

## FLOWCHART:



ROOTS OF QUADRATIC EQUATION

AIM:

To find the root of the given equation

CODING:

```
a = int(input(" Enter the first value: "))
```

```
b = int(input(" Enter the second value: "))
```

```
c = int(input(" Enter the third value: "))
```

```
d = (b * b) - (4 * a * c)
```

```
if d > 0:
```

$$x_1 = (-b + (d^{1/2})) / (2 * a)$$

$$x_2 = (-b - (d^{1/2})) / (2 * a)$$

```
print(" Roots are real and unequal ")
```

```
print(" The roots are ", x1, x2)
```

```
elif d == 0:
```

$$x_1 = x_2 = -b / (2 * a)$$

```
print(" Roots are equal ")
```

```
print(" Roots are ", x1, x2)
```

```
elif d < 0
```

$$r_p = -b / (2 * a)$$

$$i_p = ((-d)^{0.5}) / (2 * a)$$

$$x_1 = r_p + i_p$$

$$x_2 = r_p - i_p$$

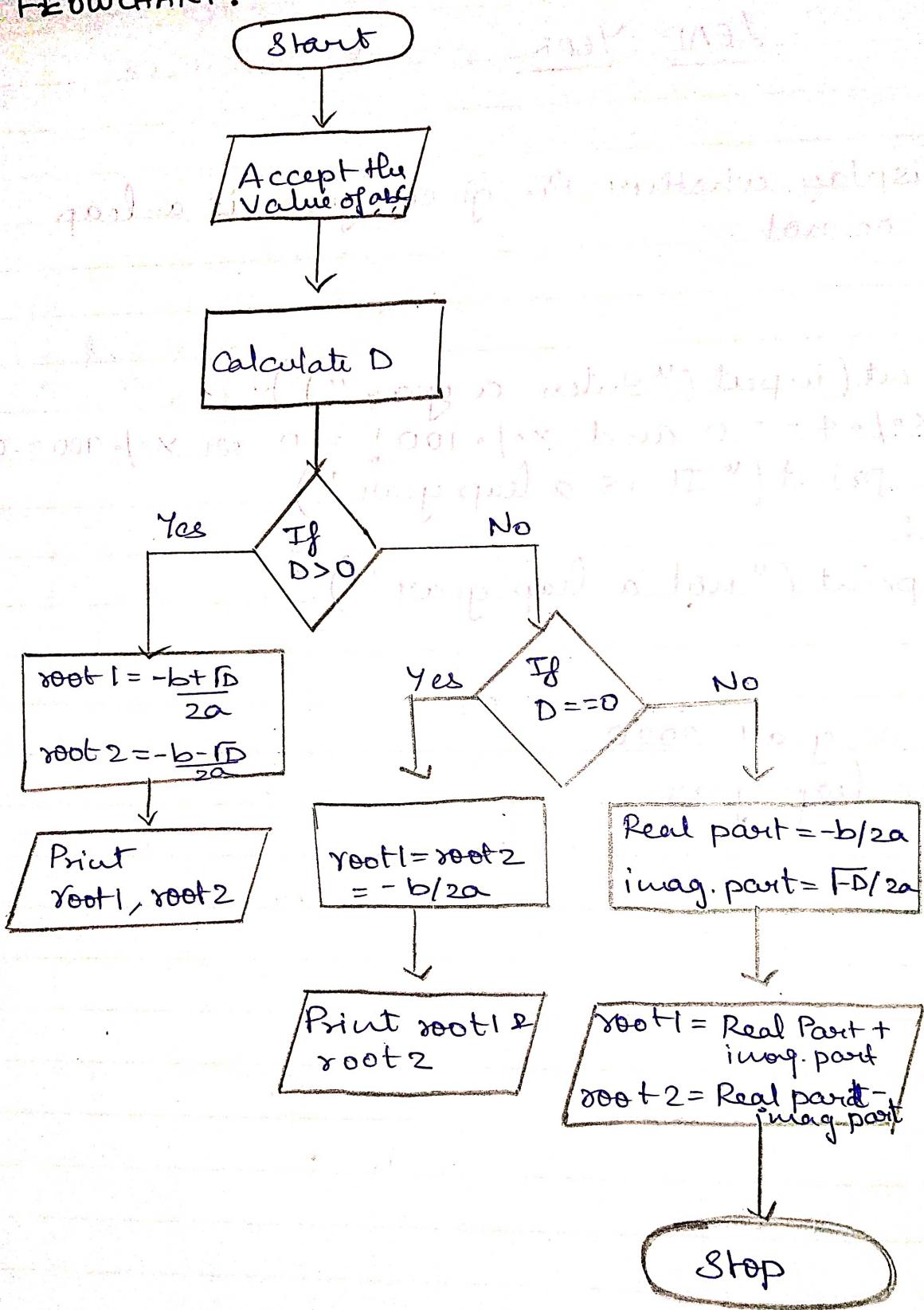
```
print(" Roots are unequal and imaginary ")
```

```
print(" The roots are ", x1, x2)
```

else:

```
print(" Invalid entry ")
```

## Flowchart:



Expt. No.....

Page No..... 17 .....

**OUTPUT:**

Enter the first value : 3

Enter the second value : 5

Enter the third value : 2

Roots are real and unequal

The roots are - 0.6666667, -1.0

Q

## SUM & MULTIPLICATION

AIM:

To find the sum of 'n' natural numbers if 'a' is pressed and find the multiplication table if 'b' is pressed.

## CODING:

```

print("1\t\t a-sum \t\t 1\n\t\t b-multiplication
      W=Add.0 = t1")
x = input("Enter an alphabet ")
n = int(input("Enter the number"))
s = 0
if x == 'a':
    for i in range(1, n+1):
        s += i
    print(s)
elif x == 'b':
    for c in range(n, n*11, n):
        print(c)

```

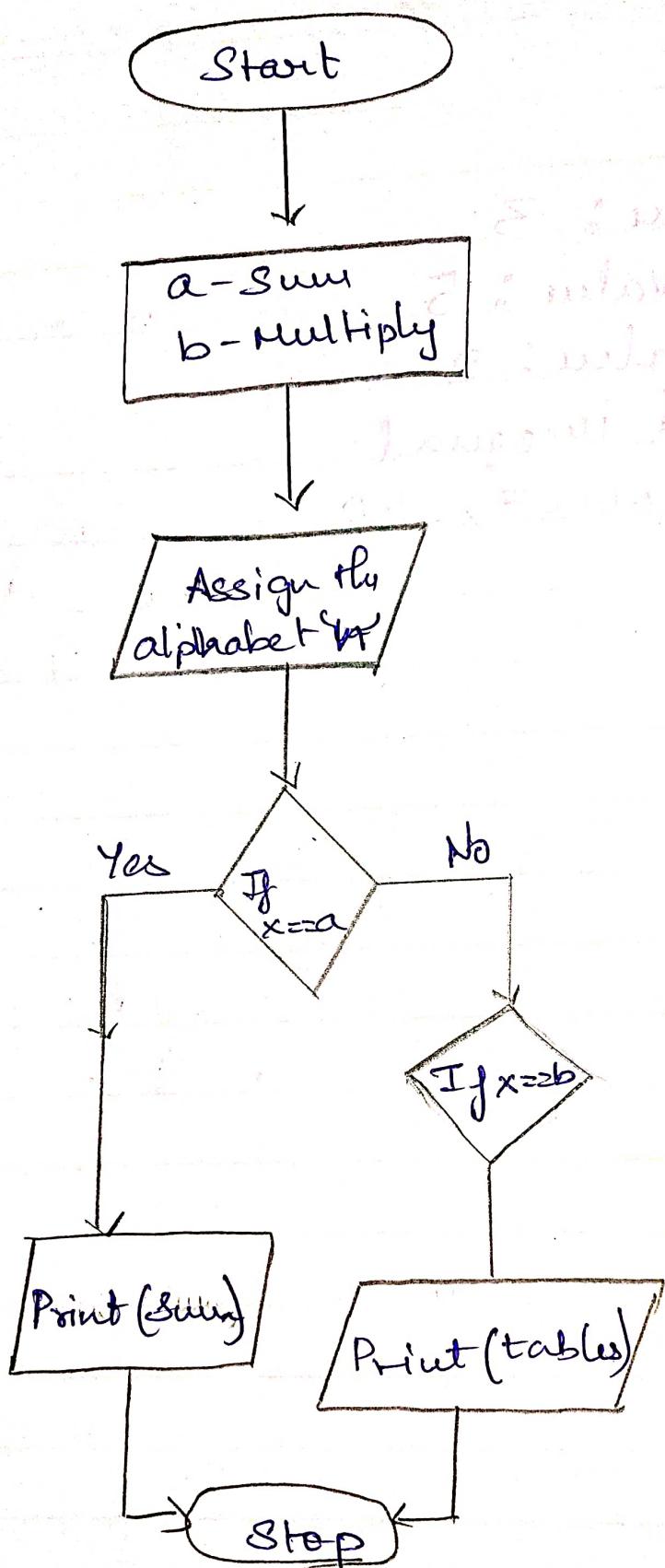
## OUTPUT:

Enter the alphabet a

Enter the numbers

15

## FLOWCHART :



## FIBONACCI SERIES

AIM:

To print the Fibonacci Series

CODING:

```
n = int(input("Enter the no. of terms"))
```

```
i, s = 0, 0
```

```
f1, f2 = -1, 1
```

```
while (i <= n):
```

```
    s = f1 + f2
```

```
    print(s)
```

```
    f1 = f2
```

```
    f2 = s
```

```
    i += 1
```

```
print("Loop over")
```

OUTPUT:

Enter the no. of terms 10

0

1

1

2

3

5

8

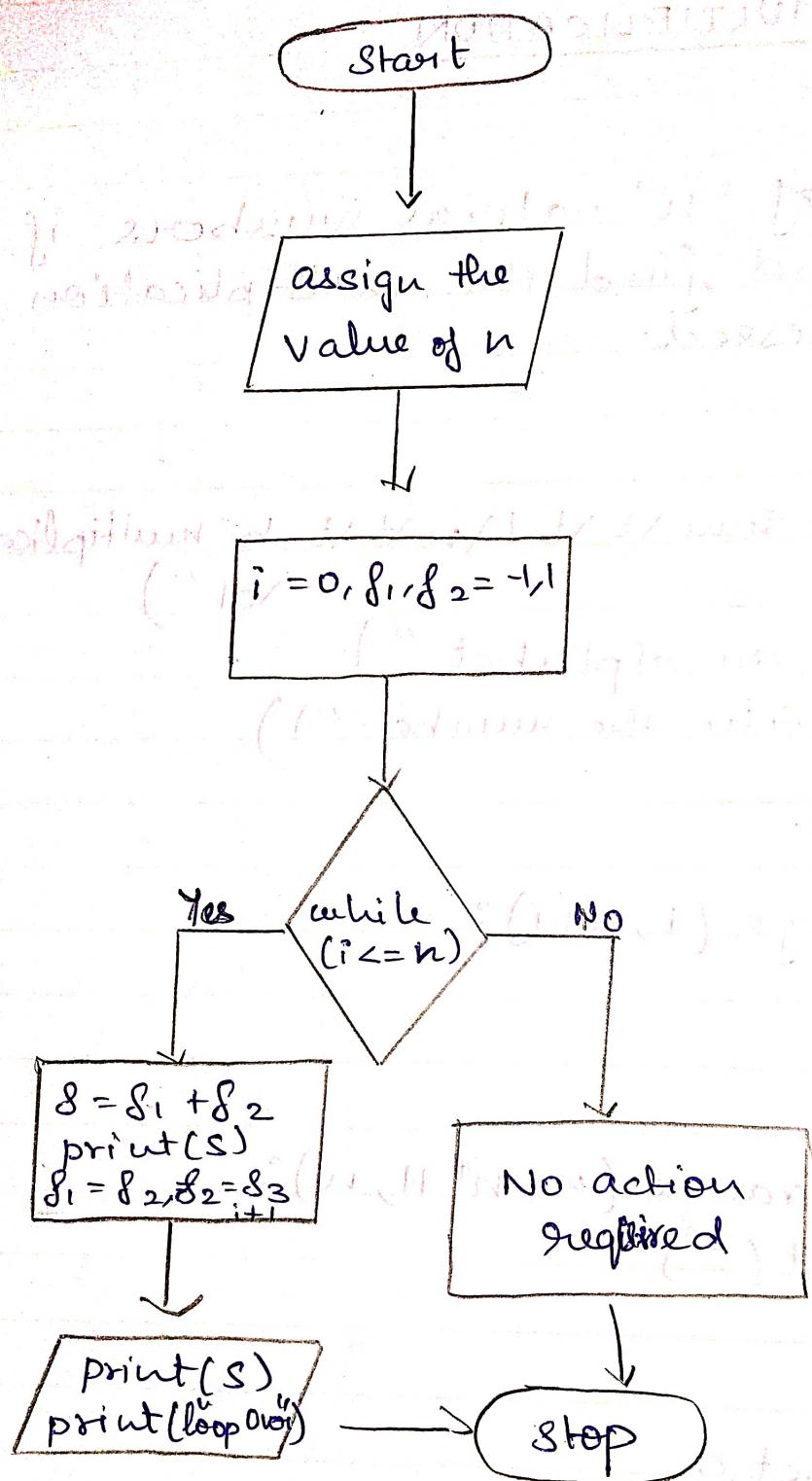
13

21

34

Loop over.

## FLOWCHART:



## PRIME AND COMPOSITE NUMBERS

AIM:

To print prime and composite numbers within a given range.

CODING:

```
a = int(input("Enter the lower limit: "))
```

```
b = int(input("Enter the upper limit: "))
```

```
c = []
```

```
d = []
```

```
for i in range(a, b+1):
```

```
    for k in range(2, i):
```

```
        if i % k == 0:
```

```
            c.append(i)
```

```
            break.
```

```
        else:
```

```
            d.append(i)
```

```
            break
```

```
print("The prime numbers are: ")
```

```
for z in d:
```

```
    print(z)
```

```
print("The Composite numbers are: ")
```

```
for x in c:
```

```
    print(x).
```



g  
✓

## ALGORITHM:

- Step 1: Start
- Step 2: accept the range from the user
- Step 3: store all prime numbers in a list  
within the given range
- Step 4: store all composite numbers in a separate  
list within the given range.
- Step 5: print the 2 lists separately.

Expt. No.....

Page No.

OUTPUT :

Enter the lower limit: 2

Enter the Upper limit: 10

the prime numbers are

2

3

5

7

The ~~not~~ Composite Numbers are

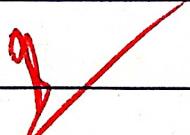
4

6

8

9

10



## SUM OF THE SERIES

### AIM:

To find the sum of the Series of  $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \pm \frac{x^n}{n!}$

### CODING:

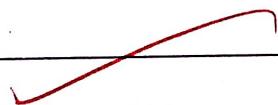
```
a = int(input("Enter the no. of terms: "))
b = int(input("Enter the number: "))
s = 0
for i in range(1, 2*a, 2):
    s = s + (-1 * -1 ** i * b) / i
s += b
print("Sum:", s)
```

### OUTPUT:

Enter the no. of terms: 3

Enter the number: 2

Sum: 5.066666666666666.



## Program to calculate sum of series

### ALGORITHM:

Step 1: Start

Step 2: accept the number and  
the number of terms  
from the user

Step 3: calculate the sum

Step 4: Display the sum.



## PALINDROME

**AIM :**

To check whether a given string is a palindrome or not using slicing and without slicing.

**CODING :**

```

print("1 1-with slicing")
print("1 2-without slicing")
st = input("Enter the string: ")
a = int(input("Enter the number from menu: "))
if a == 1:
    b = st[::-1]
    if b == st:
        print("it is a palindrome")
    else:
        print("it is not a palindrome")
elif a == 2:
    for i in range(-len(st), 0, 1):
        b += st[i]
    if st == b:
        print("palindrome")
    else:
        print("not a palindrome")

```

**OUTPUT:**

Enter the String : Kitin.

Enter the number : 1

palindrome.

## ALGORITHM:

Step 1: Start

Step 2: Display the menu.

Step 3: accept the number from the menu and the string which has to checked from the user

Step 4: check the String

Step 5: Display appropriate message.

## CAPITALISE

AIM:

To Capitalise the first letter of each word in a Sentence.

CODING:

```
s = input("Enter the string: ")
for a in s.split():
    print(a.capitalize(), end="")
```

OUTPUT:

```
Enter the string : Red is a colour
Red Is A Colour
```

## ALGORITHM:

Step 1: Start

Step 2: accept the String from  
the user

Step 3: Split the String and Capitalize  
the first word.

Step 4: Display the String.

NUMBER CONVERSION

AIM:

To convert the number into the desired conversion.

CODING:

```

print("1. bin to dec ")
print("2. dec to bin ")
print("3. Oct to dec ")
print("4. exit ")

S1, S2 = 0, 0
n = int(input("Enter the number: "))

if n == 1:
    S = int(input("Enter the number: "))
    for i in range(len(str(S)) - 1, -1, -1):
        S1 += int(str(S)[i]) * 2 ** (len(str(S)) - 1 - i)
    print(S1)

elif n == 2:
    S = int(input("Enter the number: "))
    while S != 0:
        S2 += str(S) % 2
        S = S // 2
    S2 = S2[-1::-1]
    print(S2)

elif n == 3:
    S = int(input("Enter the number: "))
    for i in range(len(str(S)) - 1, -1, -1):
        S1 += int(str(S)[i]) * 8 ** (len(str(S)) - 1 - i)
    print(S1)

elif n == 4:
    break

```

## OUTPUT:

1. bin to dec
2. dec to bin
3. Oct to dec
4. exit

enter the number : 2

enter the number : 2

10

~~next  
last presentation'~~

## ALGORITHM:

Step 1: Start

Step 2: display the menu.

Step 3: accept the number from the menu and the number to be converted from the user.

Step 4: convert the number using string/traditional method.

Step 5: Display the converted number.

## LARGEST AND SMALLEST

**AIM:**

To find the smallest, 2nd smallest, 2nd largest, largest value from a list.

**CODING:**

```

print("1 1-traditional method 1")
print("1 2-Sort() method 1")
l = eval(input("enter the list of elements :"))
x = int(input("enter the number from menu :"))
n = len(l)
if x == 1:
    for i in range(n):
        for k in range(0, n-i-1):
            if l[k] > l[k+1]:
                l[k], l[k+1] = l[k+1], l[k]
            else:
                pass
    print("sorted list:", l)
if x == 2:
    l.sort()
    print("sorted list:", l)

print("smallest:", l[0]) ✓
print("2nd smallest:", l[1])
print("2nd largest:", l[-2])
print("largest:", l[-1])

```

## OUTPUT:

1 1-traditional method

1 2-sort() method

enter the list of elements: [4, 2, 8, 3, 5, 9, 7, 0, 6, 1]

enter the number from menu: 1

sorted list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Smallest: 0

2nd Smallest: 1

2nd largest: 8

largest: 9



## ALGORITHM:

Step 1: Start

Step 2: Accept the list and number the displayed menu.

Step 3: Sort the list

Step 4: display the smallest, 2nd Smallest,  
2nd Largest, largest

VOWELS

AIM:

To find a string with the maximum no. vowels

ALGORITHM

CODING:

```
l = eval(input("enter the list of strings:"))
```

```
l1 = []
```

```
count = 0
```

```
for i in l:
```

```
    i.lower()
```

```
    for k in i
```

```
        if k in ['a', 'e', 'i', 'o', 'u']:
```

```
            count += 1
```

```
        else:
```

```
            pass
```

```
    l1.append(count)
```

```
n = l1.index(max(l1))
```

```
print("The string with maximum vowels:", l[n])
```

OUTPUT:

enter the list of strings: ['akai', 'ezeez', 'auie']

the string with maximum vowels: 'auie'.

## ALGORITHM:

Step 1: Start

Step 2: accept the list

Step 3: check every string in the list  
for vowels

Step 4: display the ~~max~~ string with the  
maximum no. of vowels .

## INTEGER SORTING

---

**AIM:**

To sort a list of integers according to user's choice.

**CODING:**

```

print(" 1 - insertion sort | \n 2 - bubble sort ")
x = int(input("enter the number from the menu:"))
l = eval(input("enter the list of elements:"))
n = len(l)
if x == 1:
    for i in range(n):
        k = l[i]
        j = i - 1
        while j >= 0 and k < l[j]:
            if l[j+1] == l[j]:
                j -= 1
            else:
                l[j+1] = k
    print(l)
if x == 2:
    for i in range(n):
        for k in range(0, n-i-1):
            if l[k] > l[k+1]:
                l[k], l[k+1] = l[k+1], l[k]
            else:
                pass
    print(l)

```

Expt. No.....

Page No.....45.....

OUTPUT:

1 1 - insertion sort )

1 2 - bubble sort )

enter the number from the menu : 1

enter the list of elements = [ 0, 9, 2, 7, 6, 3, 5, 1, 8, 4 ]  
[ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ]

✓

## ALGORITHM:

Step 1: Start

Step 2: accept the list and the number from the menu.

Step 3: display the sorted listed.

Mean  
C.M.

## AIM

To find the mean of individual elements and also the compound mean.

## CODING

```

tup = eval(input("enter the tuple of elements : ' ') )
S, C, K = 0, 0, 0
for i in tup:
    for k in i:
        S += k
    b = S / len(i)
    print("mean of ", i, " is ", b)
    C += b
    S = 0
print("The mean of means is : ", C / len(tup))

```

## OUTPUT

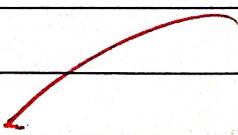
enter the tuple of elements :  $((1, 2), (3, 4.15, 5.15), (7, 8, 12, 15))$

mean of  $(1, 2)$  is 1.5

mean of  $(3, 4.15, 5.15)$  is 4.1

mean of  $(7, 8, 12, 15)$  is 10.5

mean of means is 5.366666



## ALGORITHM:

Step 1: Start

Step 2: accept the tuple from the user

Step 3: calculate the mean of the individual elements and display the mean.

Step 4: calculate the Overall mean and display the mean.

## TELEPHONE DIRECTORY.

---

**AIM**

To arrange the Telephone numbers in ascending Order by their names and also search for the details of a particular no.

**CODING**

```

d, l = {}, []
n = int(input("how many telephone users:"))
for i in range(n):
    t = int(input("enter the telephone number:"))
    n2 = input("enter the name of the user:")
    st = input("enter the state:")
    l += [n2, st]
    d[t] = l
l = []

l1, l2 = [], list(d.values())
d1 = d2 = {}

for i in l2:
    l1 += [i[0]]
l3 = list(d.keys())
for i in range(len(l1)):
    d1[l1[i]] = l3[i]

d1.sort()

l4 = l5 = []
for i in l1:
    l4 += [d1.get(i)]
for i in l4:
    print(i)

```

$ls = [d.get(i)]$

print("the ascending order:", ls)

stt = int(input("enter the number to find the details:"))

print(d.get(stt)).

#### OUTPUT:

How many telephone users : 3

enter the telephone number : 7782504

enter the name : Ravi

enter the state : Punjab

enter the telephone number : 8263110

enter the name : Shankar

enter the state : Tamil Nadu

enter the telephone number : 9855245

enter the name : Balaji

enter the state : Rajasthan.

The ascending Order : [9855245, 7782504, 8263110]

enter the number to find the details : 8263110.

[Shankar, Tamil Nadu]

✓

## ALGORITHM

Step 1 : Start

Step 2 : accept the dictionary from  
the user .

Step 3 : arrange the dictionary

Step 4 : Search for the details