Experiment No. 1

Input Specification: r,h both as float data type

Output Specification: v as float data type

Declaration: r= Radius of cone

h= height of a cone

v=Volume of a cone

Algorithm:

Step1: Start

Step2: Declare r, h, v of data type float

Step 3: Display message “Enter the radius and height”

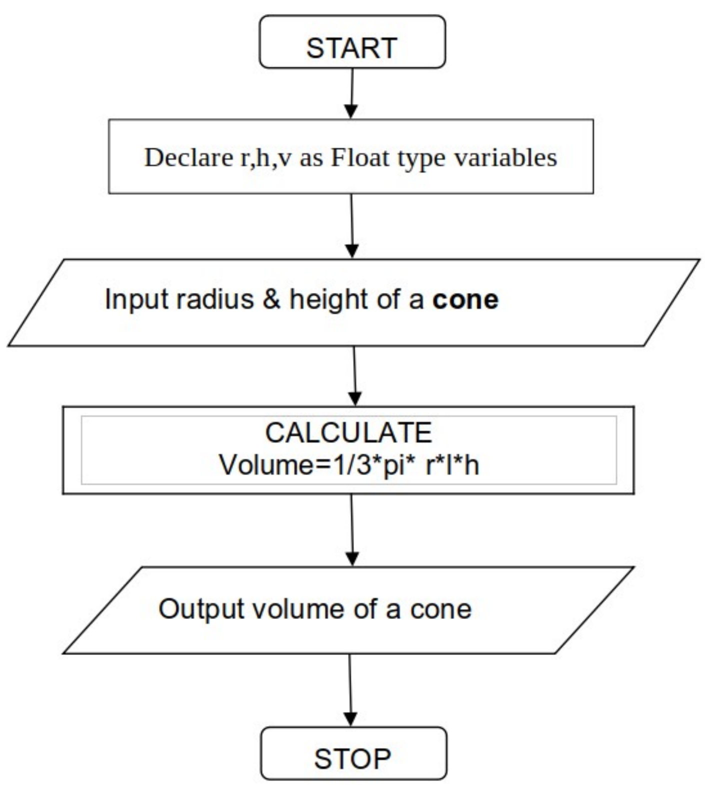
Step 4: Store them in variable r & h

Step 5: Assign value of expression (1.0/3.0)*(3.14*r*r*h) and store in v

Step 6: Print v as value of volume of cone

Step 7: stop

Flowchart:



Experiment No. 2

Input Specification: num as int data type

Output Specification: rev, sum as int data type

Declaration:

u = unit place digit as int

t = tens place digit as int

h = hundred place digit int

th = thousand place digit int

rev= reverse of four digit number

sum= sum of four digits

Algorithm:

Step 1: Start

Step 2: Declare variables u, t, h, th, num, rev of int data types

Step 3: Print Message “Enter four digit number:”

Step 4: Store input in variable num

Step 5: Check if num is four digit num then goto Step 5 otherwise
print “Error Message” and goto Step 13

Step 6: u = num % 10

Step 7: t = (num/10)% 10

Step 8: h = (num/100)% 10

Step 9: th = num/1000

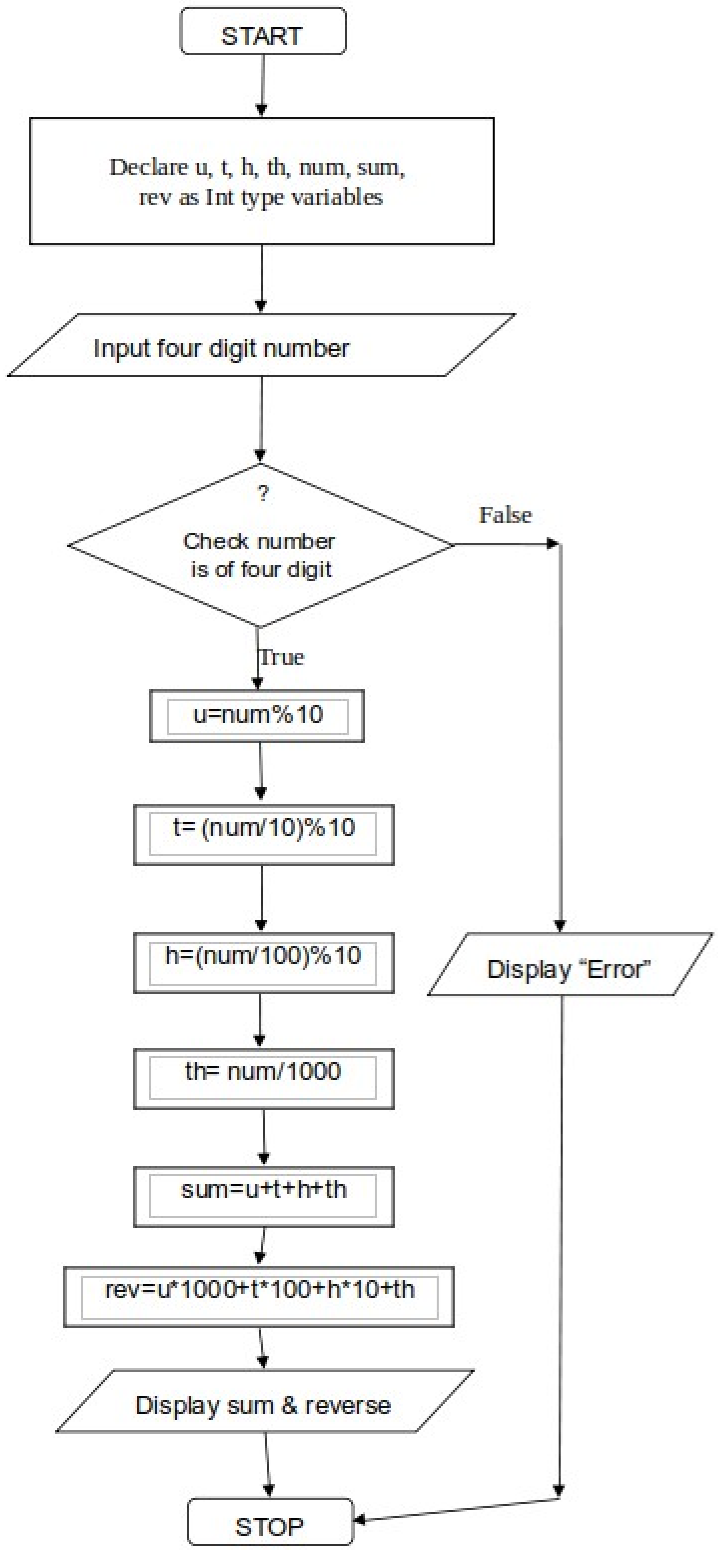
Step 10: sum= u+t+h+th

Step 11: rev= (u*1000)+(t*100)+(h*10)+th

Step 12: Print the value of rev & sum

Step 13: Stop

Flowchart:


START
Declare u, t, h, th, num, sum,
rev as Int type variables
Input four digit number
?
Check number
is of four digit
False
True
u=num%10
t= (num/10)%10
h=(num/100)%10
th= num/1000
sum=u+t+h+th
rev=u*1000+t*100+h*10+th
Display sum & reverse
Display "Error"
STOP

Experiment No. 3

Input Specification: year as unsigned int data type

Output Specification: message specifying if given year is leap or not

Declaration:

year = to store input year

Algorithm:

Step 1: Start

Step 2: Declare year as unsinged int data type

Step 3: Print message “Input year to test”

Step 4: Store input in variable year

Step 5: Check if year % 100 == 0

Then goto Step 6

Otherwise goto Step 7

Step 6: Check if year % 400 == 0

Then Print message “Year is Leap” goto 8

Otherwise Print message “Year is not Leap” goto 8

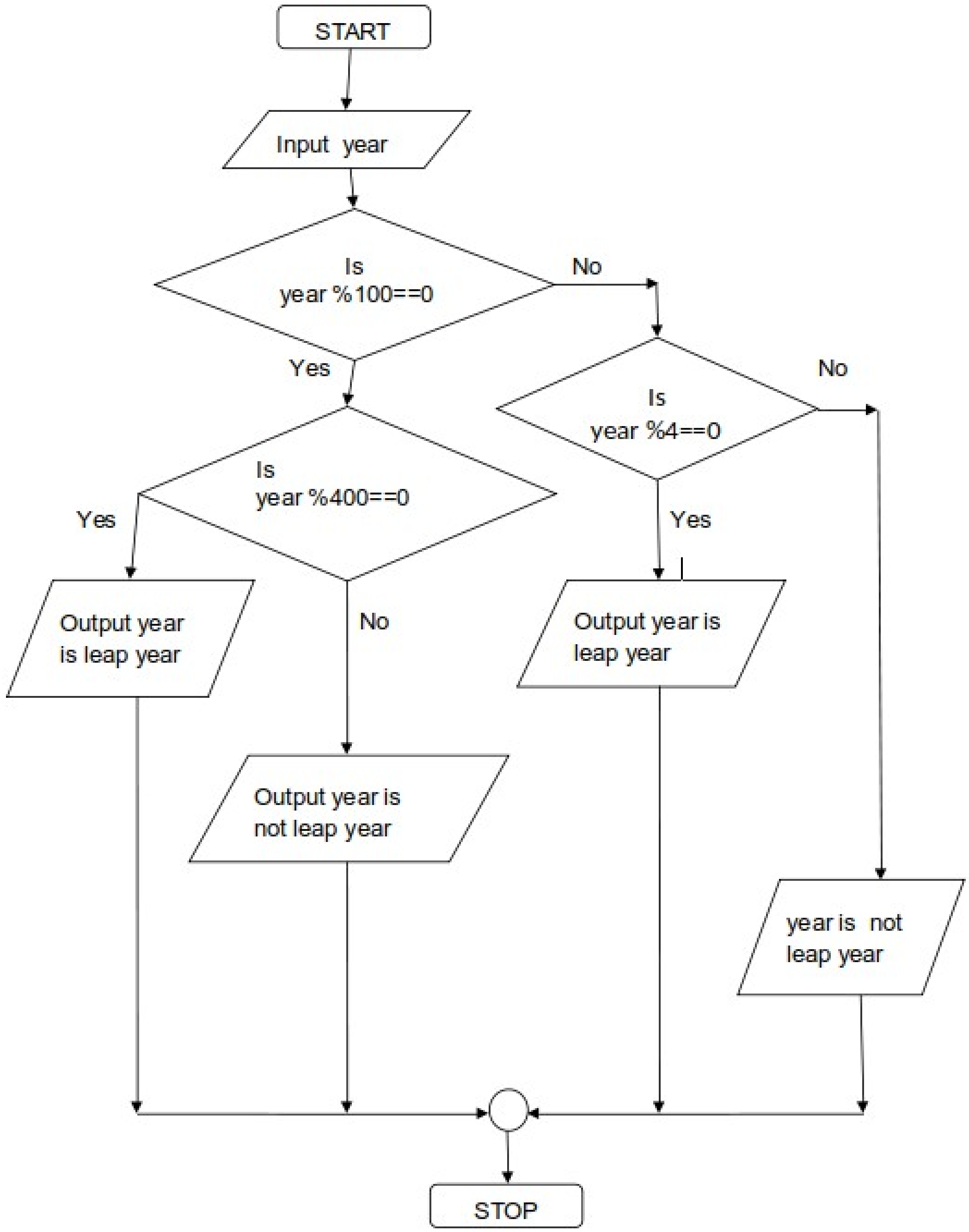
Step 7: Check if year % 4 == 0

Then Print message “Year is Leap” goto 8

Otherwise Print message “Year is not Leap” goto 8

Step 8: Stop

Flowchart:


START
Input year
Is
year %100==0
No
Yes
Is
year %4==0
No
Is
year %400==0
Yes
Yes
Output year
is leap year
No
Output year is
leap year
Output year is
not leap year
year is not
leap year
STOP

Practical No. 4

Input Specification: n as int data type

Output Specification: Print message if The number is Armstrong or not

Declaration:

n = input three digit number

sum = to store cubed sum of all three digits

u = to store digit at units place

t = to store digit at tens place

h = to store digit at hundred place

Algorithm:

Step 1: Start

Step 2: Define cube(m) as m*m*m

Step 2: Declare n,u,t,h as int data type variables and declare sum as int data type variable and initialize it to zero

Step 3: Print Message “Input Three Digit Number”

Step 4: Store input in variable n

Step 5: u= n%10

Step 6: t= (n/10)%10

Step 7: h= (n/100)%10

Step 8: sum=cube(u)+cube(t)+cube(h)

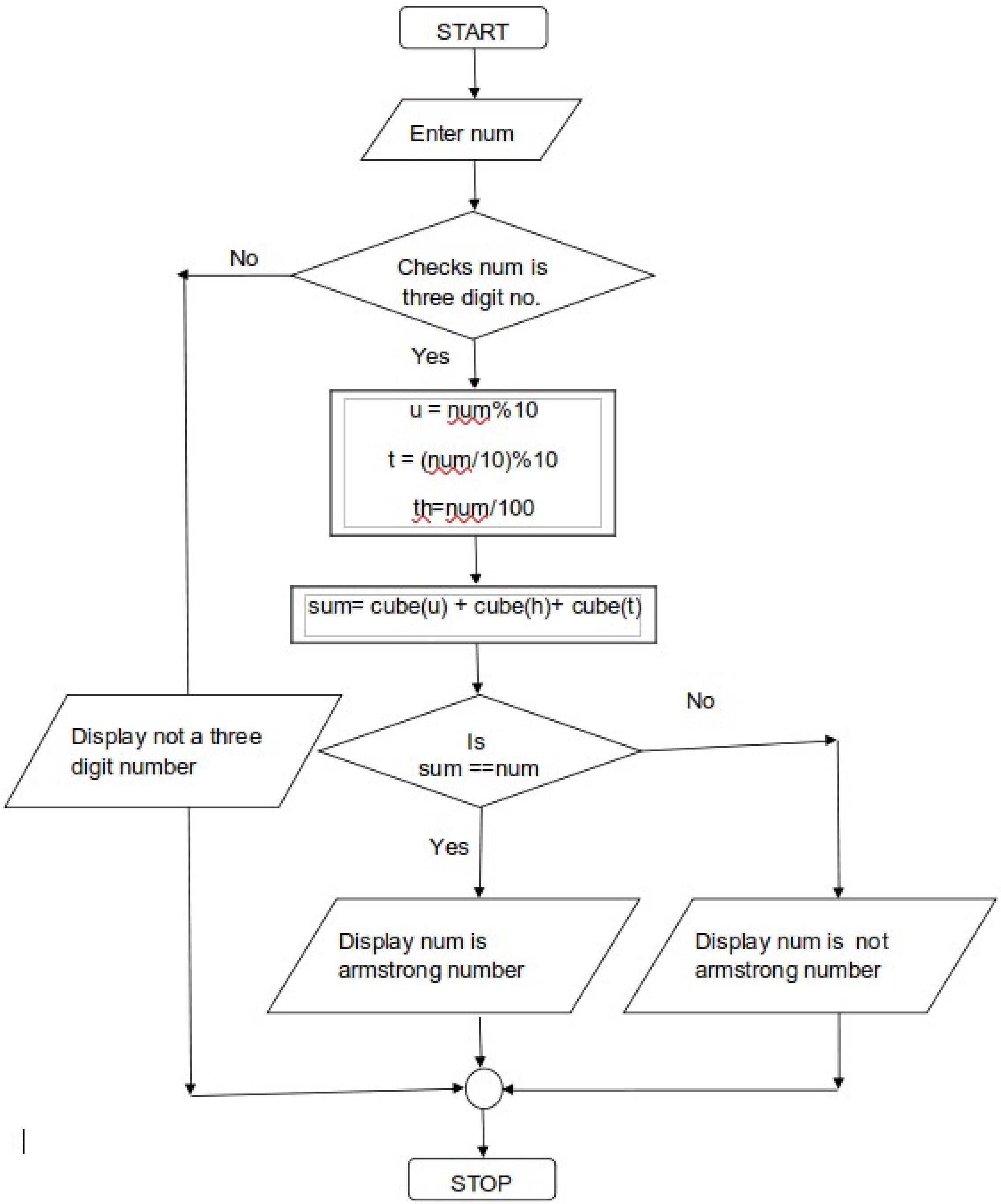
Step 9: check if sum == n

Then Print “Number is Armstrong “ goto step 10.

Otherwise Print “Number is not Armstrong” goto step 10.

Step 10: Stop

Flowchart:



START
Enter num
Checks num is three digit no.
No
Yes
u = num%10
t = (num/10)%10
th=num/100
sum= cube(u) + cube(h)+ cube(t)
No
Is sum ==num
Display not a three digit number
Yes
Display num is armstrong number
Display num is not armstrong number
STOP

Experiment No. 5

Input Specification: a, b, c as int data type

Output Specification: r1 and r1 as float data type

Declaration:

a, b, c = to store coefficients of quadratic equations as int

r1, r2 = to store roots of quadratic equations as float

z = to store b*b - 4*a*c

Algorithm:

Step1: Start

Step 2: Declare a,b, c as int and r1, r2,z as float

Step 3: Print message to input a, b, c

Step 3: store values input in a, b, c

Step 4: if a == 0 then display “Equation is not quadratic” goto Step 8

else goto Step 5

Step 5: z=(b*b)-(4*a*c)

Step 6:if z == 0 yes

then 6.1 Print "Roots are real & equal.....\n"

6.2 r1 = r2 = -b/(2*a)

6.3 Print roots as r1 and r2

6.4 goto Step 8

else goto step 7

Step 7: if z > 0 "Roots are real & distinct.....\n"

then 7.1 Print "Roots are real & distinct.....\n"

7.2 r1=(float)(-b+sqrt(z))/(2*a)

7.3 r2=(float)(-b-sqrt(z))/(2*a)

7.4 Print roots as r1 and r2

7.5 goto Step 8

else 7.1 Print "Roots are imaginary....\n"

7.2 r1= -(float)b/(2*a)

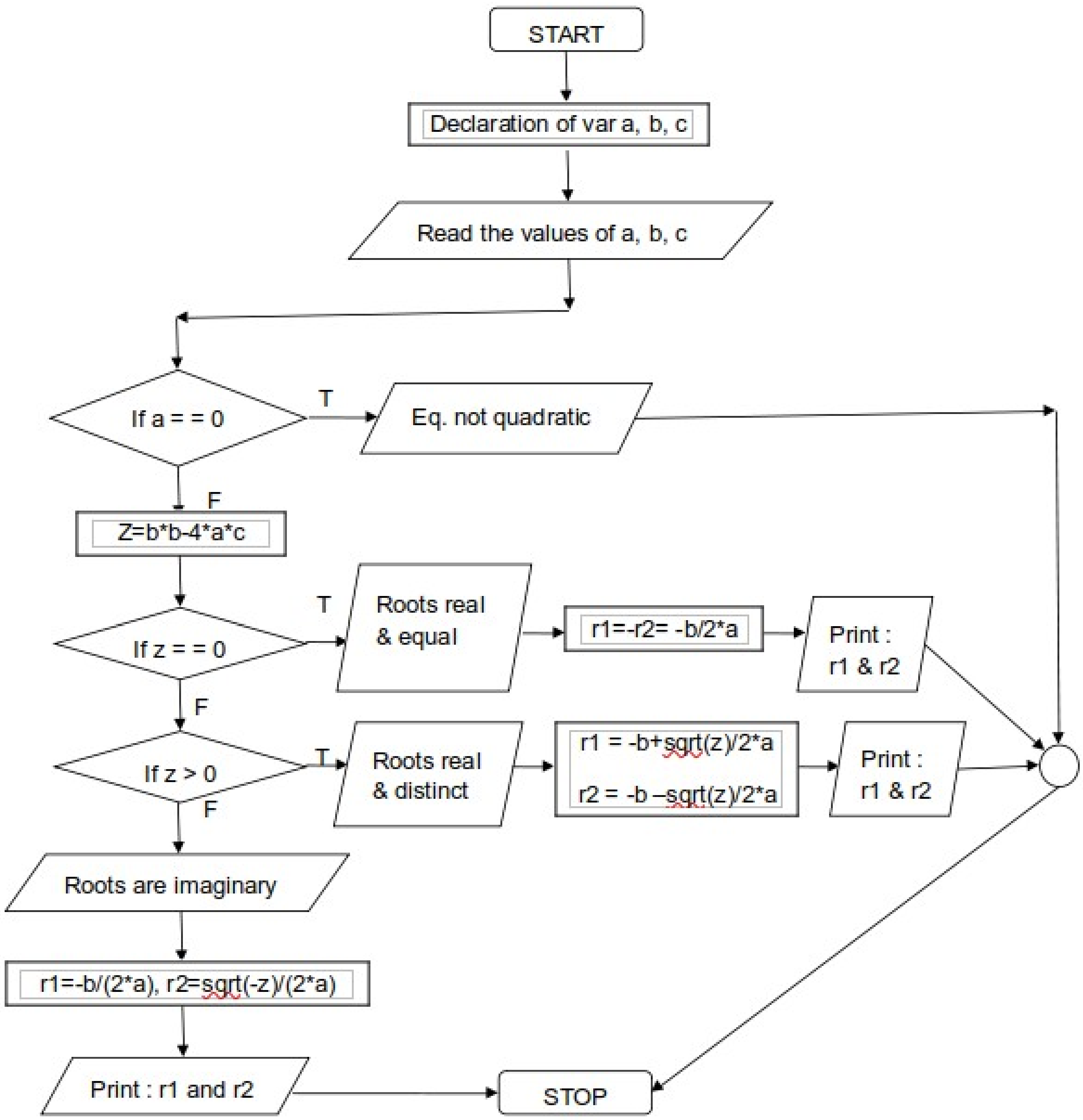
7.3 r2= sqrt(-z)/(2*a)

7.4 Print both the roots using r1 and r2

7.5 goto Step 8

Step 8: Stop.

Flowchart:


START
Declaration of var a, b, c
Read the values of a, b, c
If a = = 0
T
Eq. not quadratic
F
Z=b*b-4*a*c
If z = = 0
T
Roots real & equal
r1=-r2= -b/2*a
Print : r1 & r2
F
If z > 0
T
Roots real & distinct
r1 = -b+sqrt(z)/2*a
r2 = -b –sqrt(z)/2*a
Print : r1 & r2
F
Roots are imaginary
r1=-b/(2*a), r2=sqrt(-z)/(2*a)
Print : r1 and r2
STOP