TYPE-B:INTRODUCTION TO PROBLEM SOLVING:CH-4

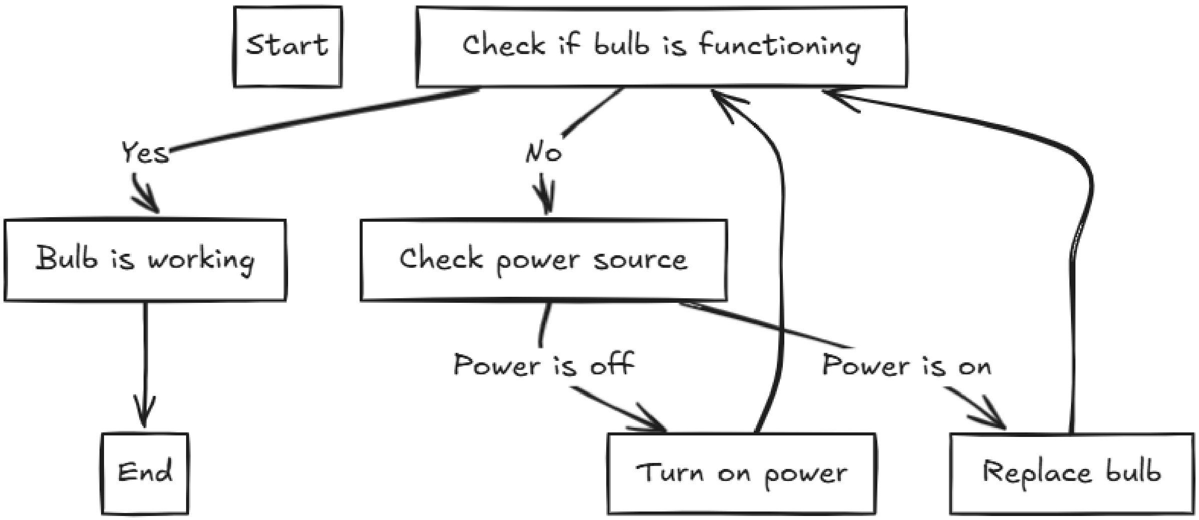
1) Write an algorithm to find square of two numbers.

sol:

```
Step 1: Start
Step 2: Input the first number and store it in num1
Step 3: Input the second number and store it in num2
Step 4: Calculate the square of the first number: square1 = num1 × num1
Step 5: Calculate the square of the second number: square2 = num2 × num2
Step 6: Display the value of square1
Step 7: Display the value of square2
Step 8: End
```

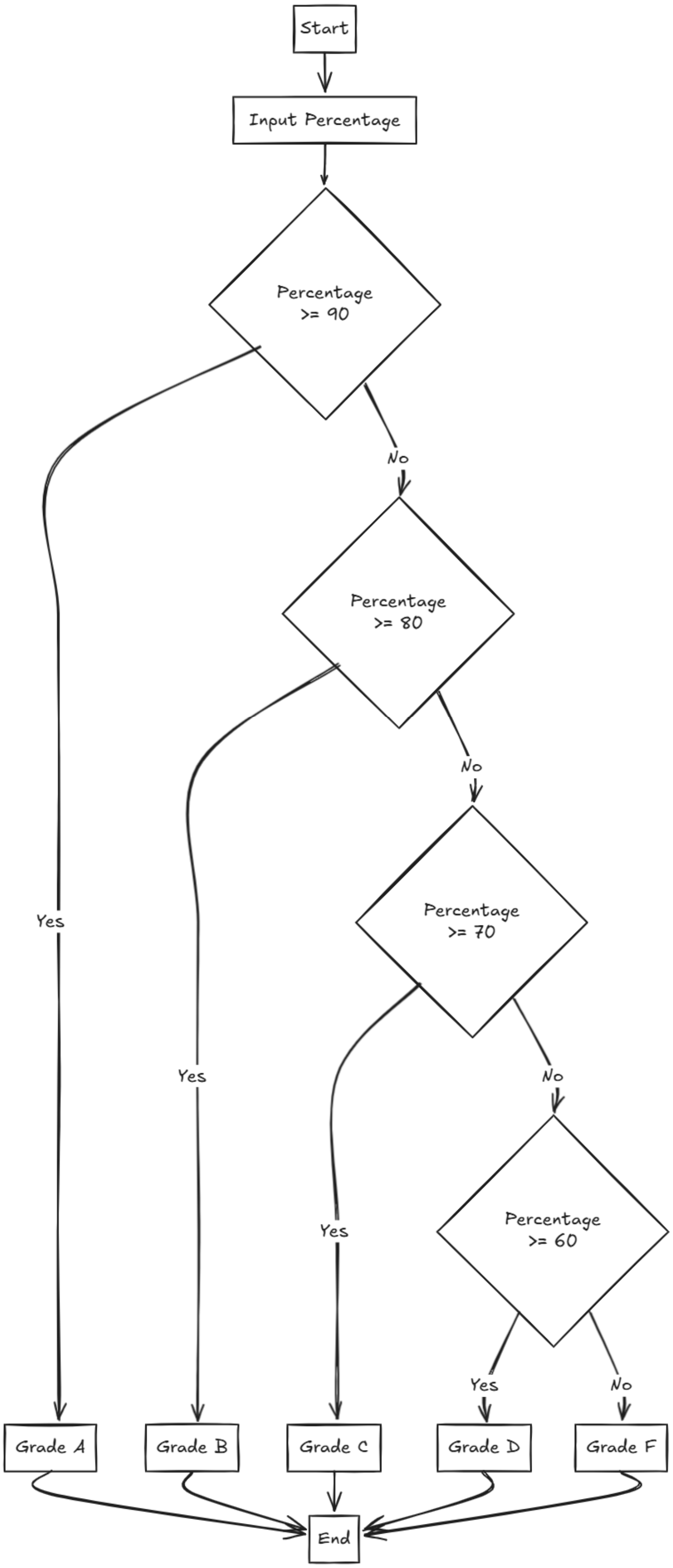
2) draw the flowchart to solve the problem of non functioning bulb.

sol:



3) draw a flowchart to calculate grade from mark percentage.

sol:



4) write an alogorithm to double a

number in two different ways
(i) n+n
(ii) 2 x n

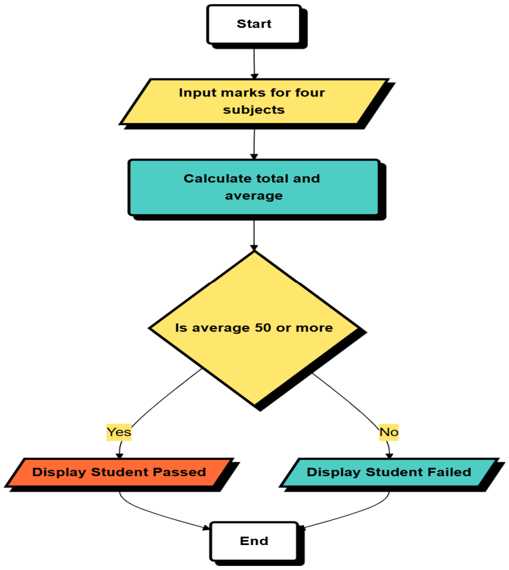
sol:

```
Step 1: Start
Step 2: Input a number and store it in n
Step 3: Method 1: Calculate double using n + n → double1 = n + n
Step 4: Method 2: Calculate double using 2 × n → double2 = 2 × n
Step 5: Display the value of double1
Step 6: Display the value of double2
Step 7: End
```

5) Write an algorithm and flowchart to determine if a student passed the exam or not.(note there are 4 subjects papers and passing average is 50 or more)

sol:

```
Step 1: Start
Step 2: Input marks for Subject 1 → marks1
Step 3: Input marks for Subject 2 → marks2
Step 4: Input marks for Subject 3 → marks3
Step 5: Input marks for Subject 4 →           marks4
Step 6: Calculate total marks → total = marks1 + marks2 + marks3 + marks4
Step 7: Calculate average marks → average = total ÷ 4
Step 8: Is average ≥ 50?
     Yes → Step 9
     No  → Step 10
Step 9: Display "Student Passed" → Go to Step 11
Step 10: Display "Student Failed" → Go to Step 11
Step 11: End
```



6) Write the peusdo code for the

following algorithm

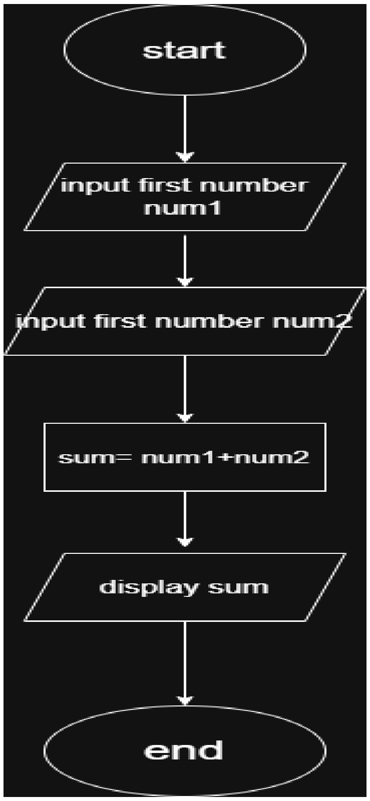
```
a) START
   READ X, Y, Z
   IF X > Y THEN
      IF X > Z THEN
         PRINT X IS THE LARGEST NUMBER
      ELSE
         PRINT Z IS THE LARGEST NUMBER
   ELSE
      IF Y > Z THEN
         PRINT Y IS THE LARGEST NUMBER
      ELSE
         PRINT Z IS THE LARGEST NUMBER
   STOP
```

```
b)START
   SET C = 0
   SET S = 0
   WHILE C < 5
      INPUT N
      S = S + N
      C = C + 1
   AVG = S / 5
   PRINT AVG
   STOP
```

7) Write an algorithm to display the sum of two numbers entered by user, using both pseudocode and flowchart.

sol:

```
Step 1: Start
Step 2: Input first number → num1
Step 3: Input second number → num2
Step 4: Calculate sum → sum = num1 + num2
Step 5: Display sum
Step 6: End
```



8) To find area and perimeter of a rectangle

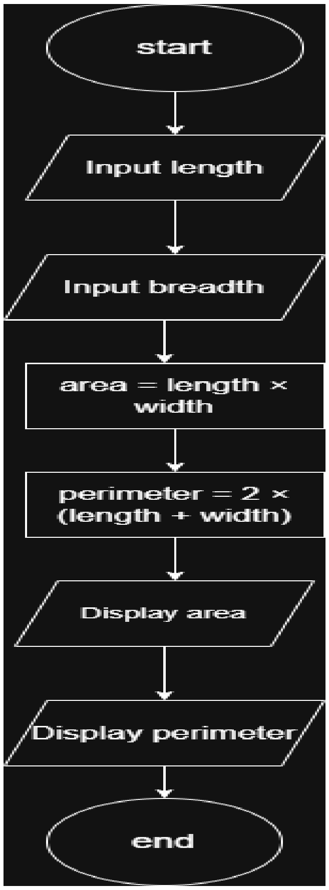
sol:

algorithm

```
  Step 1: Start
  Step 2: Input length of rectangle → length
  Step 3: Input width of rectangle → width
  Step 4: Calculate area → area = length × width
  Step 5: Calculate perimeter → perimeter = 2 × (length + width)
  Step 6: Display area
  Step 7: Display perimeter
  Step 8: End
```

pseudocode

```
   START
    INPUT length
    INPUT width
    area ← length * width
    perimeter ← 2 * (length + width)
    PRINT "Area of rectangle is:", area
    PRINT "Perimeter of rectangle is:", perimeter
  END
```



9) To calculate the area and circumference of the circle

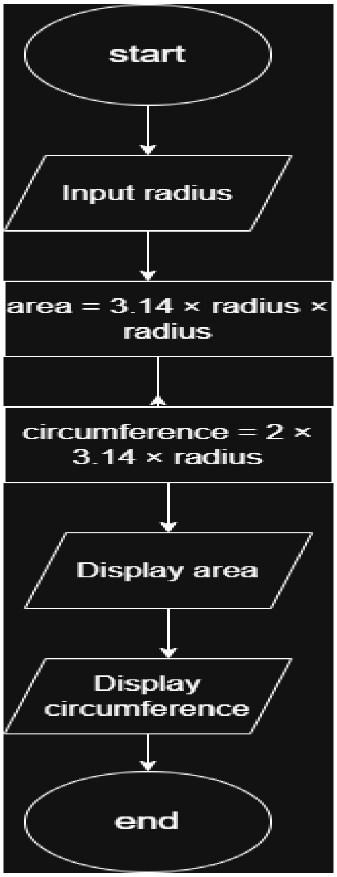
sol:

algorithm

```
  Step 1: Start
  Step 2: Input the radius of the circle → radius
  Step 3: Calculate area → area = 3.14 × radius × radius
  Step 4: Calculate circumference → circumference = 2 × 3.14 × radius
  Step 5: Display area
  Step 6: Display circumference
  Step 7: End
```

pseudocode

```
  START
    INPUT radius
    area ← 3.14 * radius * radius
    circumference ← 2 * 3.14 * radius
    PRINT "Area of the circle is:", area
    PRINT "Circumference of the circle is:", circumference
  END
```


start
Input radius
area = 3.14 × radius × radius
circumference = 2 × 3.14 × radius
Display area
Display circumference
end

10) To calculate the simple interest

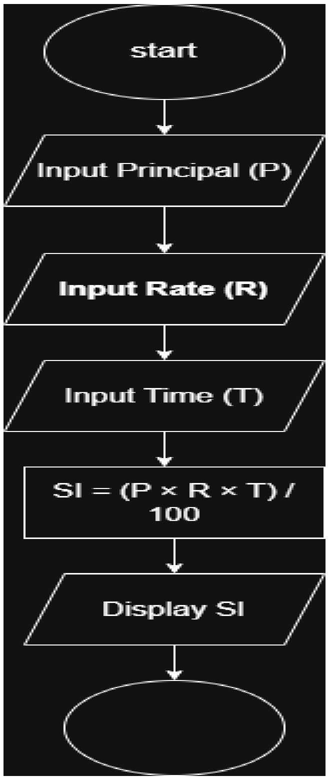
sol:

algorithm:

```
Step 1: Start
Step 2: Input Principal amount → P
Step 3: Input Rate of interest → R
Step 4: Input Time period in years → T
Step 5: Calculate simple interest → SI = (P × R × T) / 100
Step 6: Display SI
Step 7: End
```

pseudocode:

```
START
  INPUT P
  INPUT R
  INPUT T
  SI ← (P * R * T) / 100
  PRINT "Simple Interest is:", SI
END
```



11) To check whether a year is aleap year or not

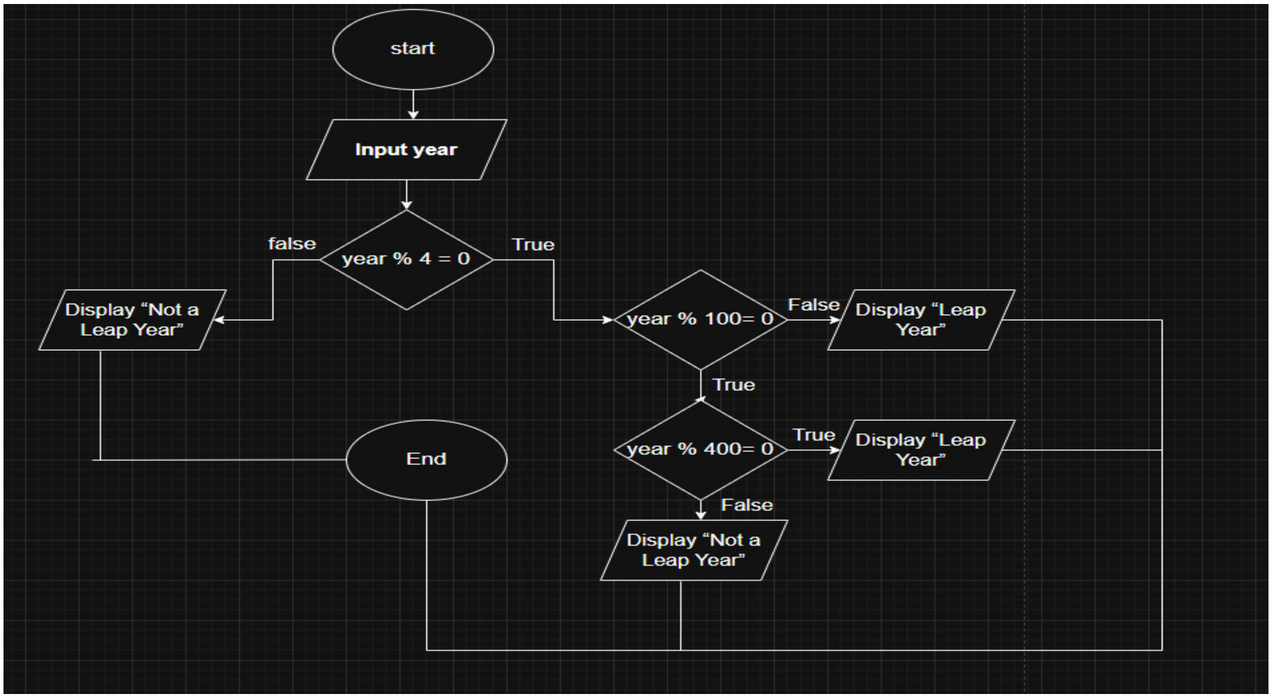
sol:

algoritm:

```
Step 1: Start
Step 2: Input the year → year
Step 3: If year is divisible by 4 then
          If year is divisible by 100 then
            If year is divisible by 400 then
              Display "Leap Year"
            Else
              Display "Not a Leap Year"
          Else
            Display "Leap Year"
        Else
          Display "Not a Leap Year"
Step 4: End
```

pseudocode:

```
START
  INPUT year
  IF (year MOD 4 = 0) THEN
    IF (year MOD 100 = 0) THEN
      IF (year MOD 400 = 0) THEN
        PRINT "Leap Year"
      ELSE
        PRINT "Not a Leap Year"
    ELSE
      PRINT "Leap Year"
  ELSE
    PRINT "Not a Leap Year"
END
```

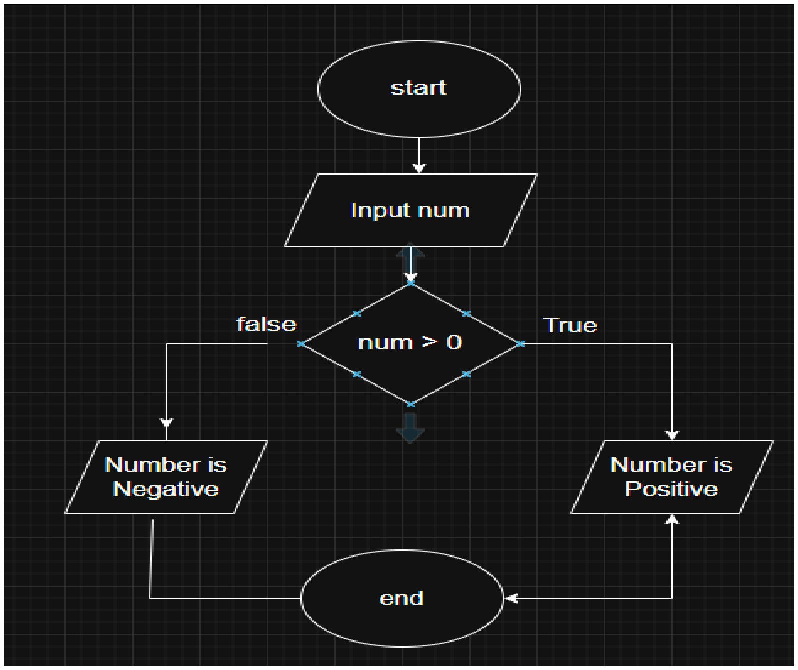


12) To check if a number is positive or negative

sol:

algorithm:

```
Step 1: Start
Step 2: Input a number → num
Step 3: If num > 0 then
            Display "Number is Positive"
        Else if num < 0 then
            Display "Number is Negative"
        Else
            Display "Number is Zero"
Step 4: End
```

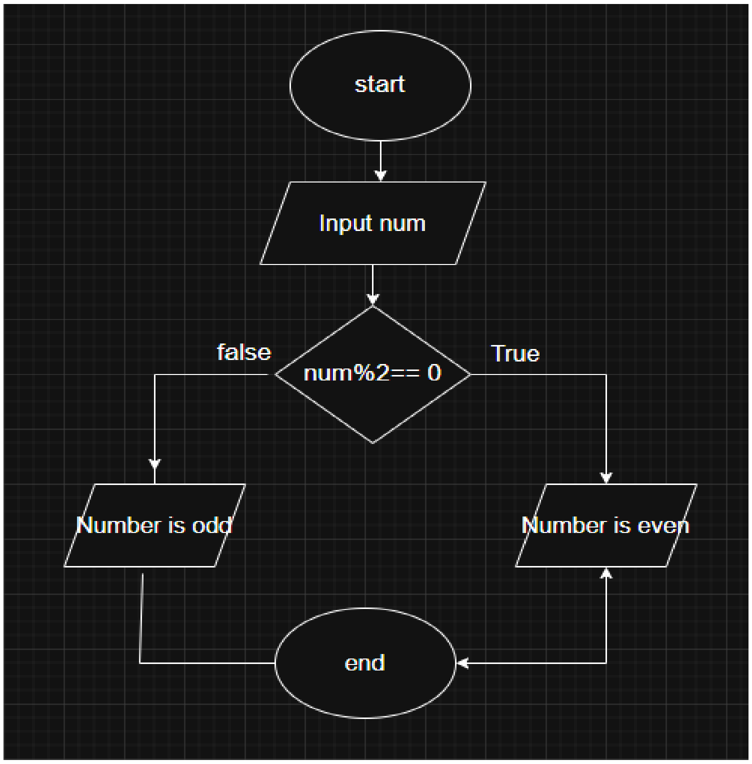


pseudocode:

```
START
  INPUT num
  IF num > 0 THEN
    PRINT "Number is Positive"
  ELSE IF num < 0 THEN
    PRINT "Number is Negative"
  ELSE
```

```
      PRINT "Number is Zero"
  END
```

13) To check the number odd or even



sol:

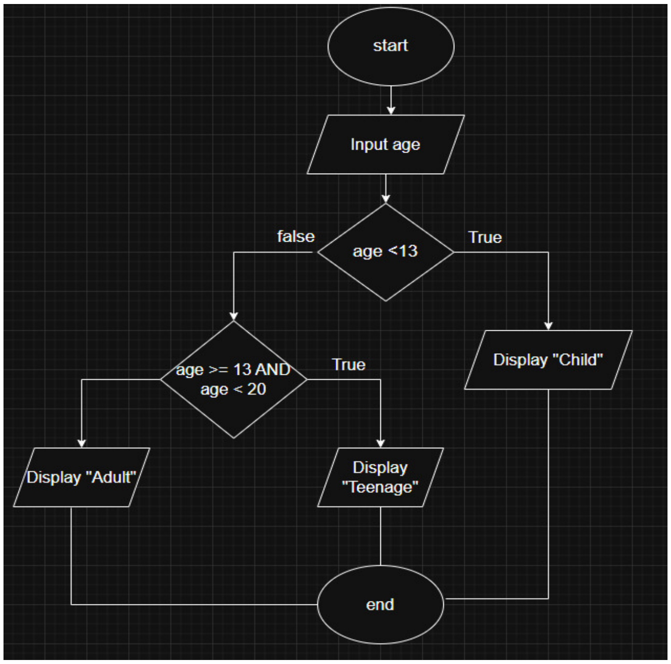
algorithm:

```
  Step 1: Start
  Step 2: Input a number → num
  Step 3: If num MOD 2 = 0 then
              Display "Number is Even"
            Else
              Display "Number is Odd"
  Step 4: End
```

pseudocode:

```
  START
    INPUT num
    IF num MOD 2 = 0 THEN
      PRINT "Number is Even"
    ELSE
      PRINT "Number is Odd"
  END
```

14) To categorise a person as either child (<13), teenage (>=13 but <20) or adult (>=20), based on given age.



sol:

algorithm:

```
  Step 1: Start
  Step 2: Input age → age
  Step 3: If age < 13 then
              Display "Child"
            Else if age >= 13 AND age < 20 then
              Display "Teenage"
            Else
              Display "Adult"
  Step 4: End
```

pseudocode:

```
  START
    INPUT age
    IF age < 13 THEN
```

```
      PRINT "Child"
    ELSE IF age >= 13 AND age < 20 THEN
      PRINT "Teenage"
    ELSE
      PRINT "Adult"
END
```

15) To print all natural numbers up to n

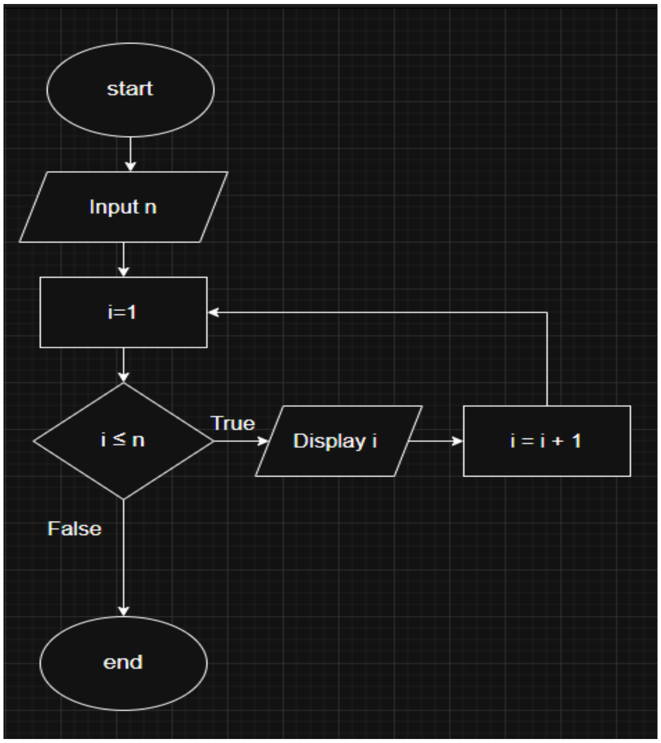
sol:

algorithm:

```
Step 1: Start
Step 2: Input n
Step 3: Initialize counter → i = 1
Step 4: While i ≤ n do
        Display i
        Increment i → i = i + 1
Step 5: End
```

pseudocode:

```
START
  INPUT n
  i ← 1
  WHILE i ≤ n DO
    PRINT i
    i ← i + 1
END
```



16) To print n odd numbers

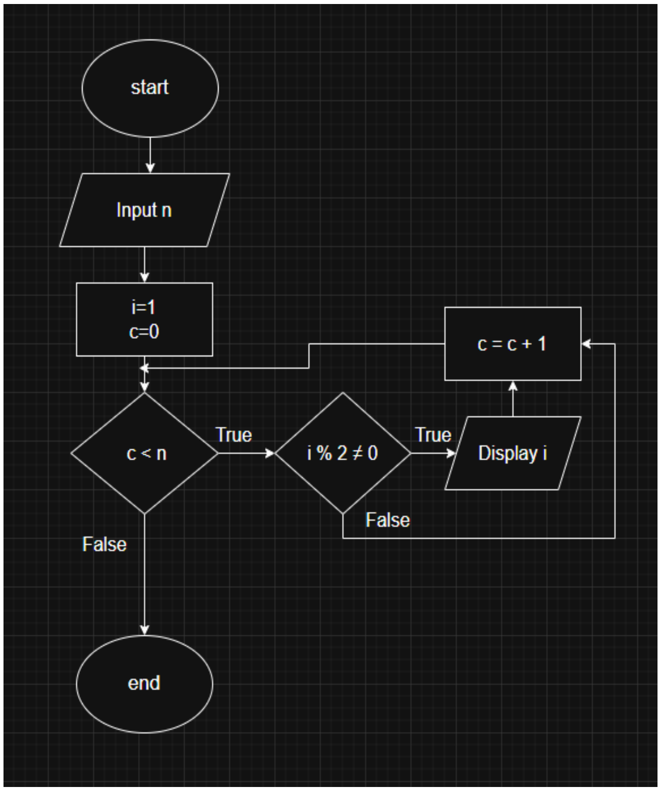
sol:

algorithm:

```
Step 1: Start
Step 2: Input n
Step 3: Initialize counter → i = 1
Step 4: Initialize count → c = 0
Step 5: While c < n do
        If i MOD 2 ≠ 0 then
          Display i
          Increment count → c = c + 1
        Increment i → i = i + 1
Step 6: End
```

pseudocode:

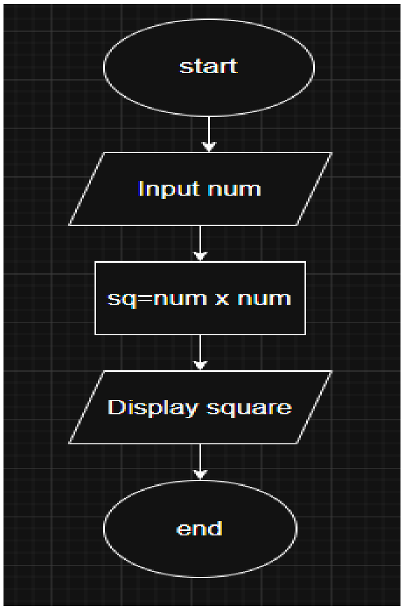
```
START
  INPUT n
  i ← 1
```



```
  c ← 0
  WHILE c < n DO
    IF i MOD 2 ≠ 0 THEN
      PRINT i
      c ← c + 1
    i ← i + 1
END
```

17) To print square of a number

sol:



algorithm:

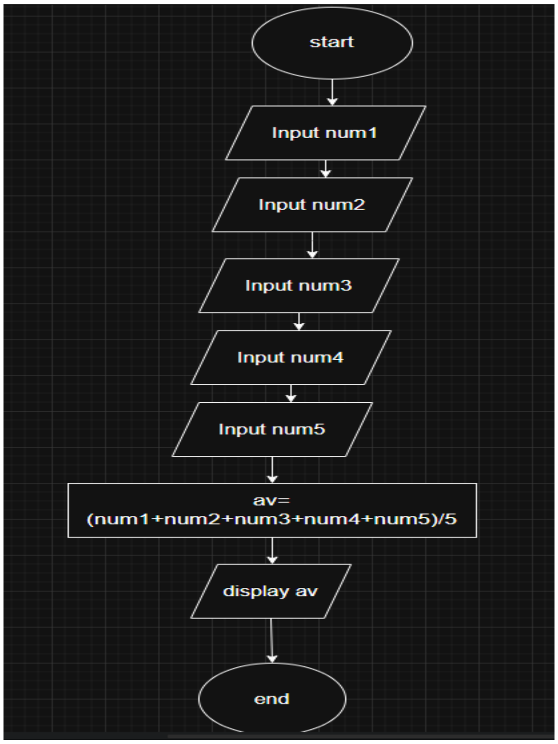
Step 1: Start
Step 2: Input a number → num
Step 3: Calculate square → square = num × num
Step 4: Display square
Step 5: End

pseudocode:

```
START
  INPUT num
  square ← num * num
  PRINT "Square of the number is:", square
END
```

18) To accept 5 numbers and find their average

sol:



algorithm:

Step 1: Start
Step 2: Input number1 → num1
Step 3: Input number2 → num2
Step 4: Input number3 → num3
Step 5: Input number4 → num4
Step 6: Input number5 → num5
Step 7: Calculate sum →
sum = num1 + num2 + num3 + num4 + num5
Step 8: Calculate average → average = sum / 5
Step 9: Display average
Step 10: End

pseudocode:

```
START
  INPUT num1
  INPUT num2
  INPUT num3
  INPUT num4
  INPUT num5
```

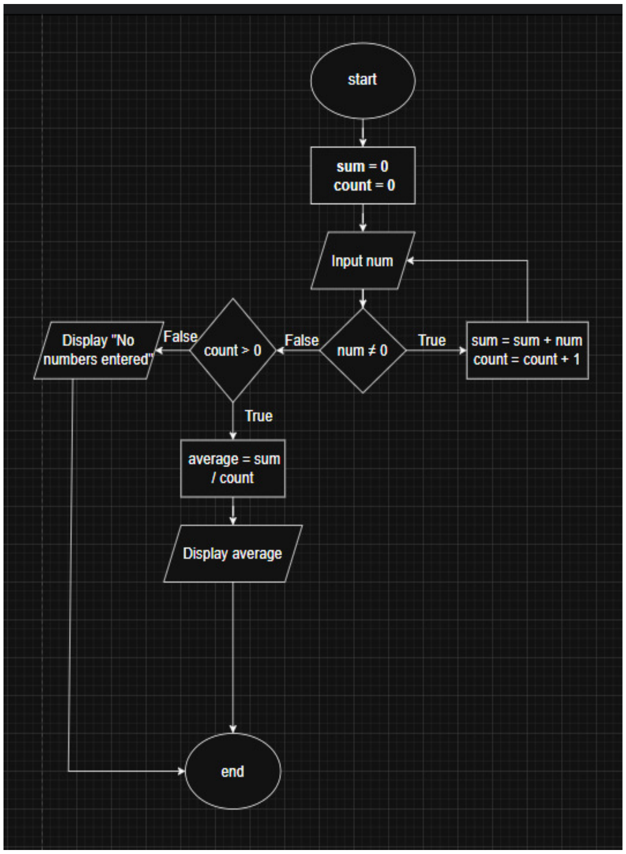
```
    sum ← num1 + num2 + num3 + num4 + num5
    average ← sum / 5
    PRINT "Average of the 5 numbers is:", average
  END
```

19) To accept the numbers till the user enters 0 and then find their average

sol:

algorithm:

```
  Step 1: Start
  Step 2: Initialize sum ← 0, count ← 0
  Step 3: Input number → num
  Step 4: While num ≠ 0
              sum ← sum + num
              count ← count + 1
              Input next number → num
  Step 5: If count > 0 then
              average ← sum / count
              Display average
          Else
              Display "No numbers entered"
  Step 6: End
```



pseudocode:

```
  START
    sum ← 0
    count ← 0
    INPUT num
    WHILE num ≠ 0 DO
      sum ← sum + num
      count ← count + 1
      INPUT num
    END WHILE
    IF count > 0 THEN
      average ← sum / count
      PRINT "Average is:", average
    ELSE
      PRINT "No numbers entered"
  END
```

20) To print square of first n numbers

sol:

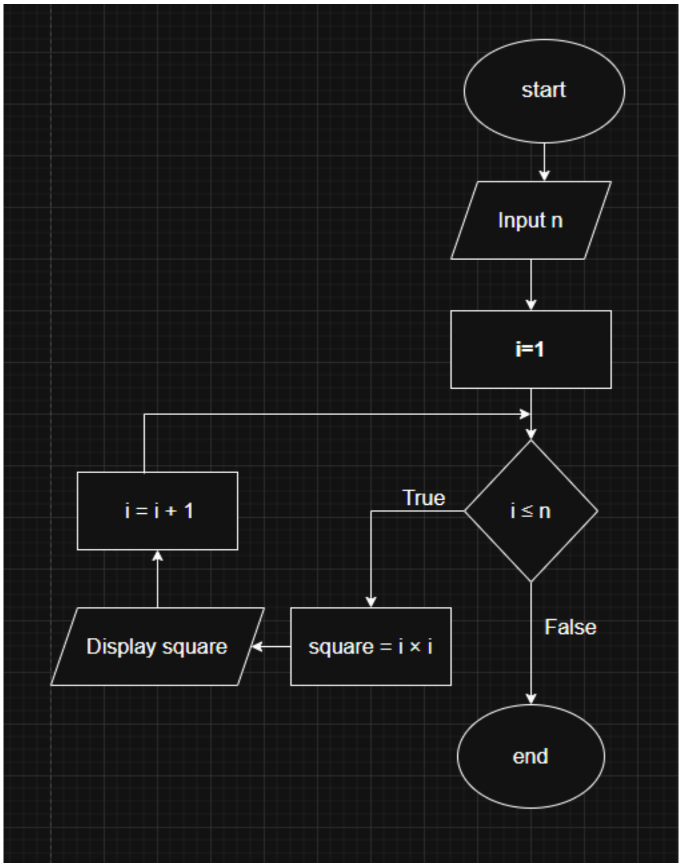
algorithm:

```
  Step 1: Start
```

```
Step 2: Input n
Step 3: Initialize i ← 1
Step 4: While i ≤ n do
          square ← i × i
          Display square
          i ← i + 1
Step 5: End
```

psuedocode:

```
START
  INPUT n
  i ← 1
  WHILE i ≤ n DO
    square ← i * i
    PRINT "Square of", i, "is", square
    i ← i + 1
END
```



21) To print cube of a number

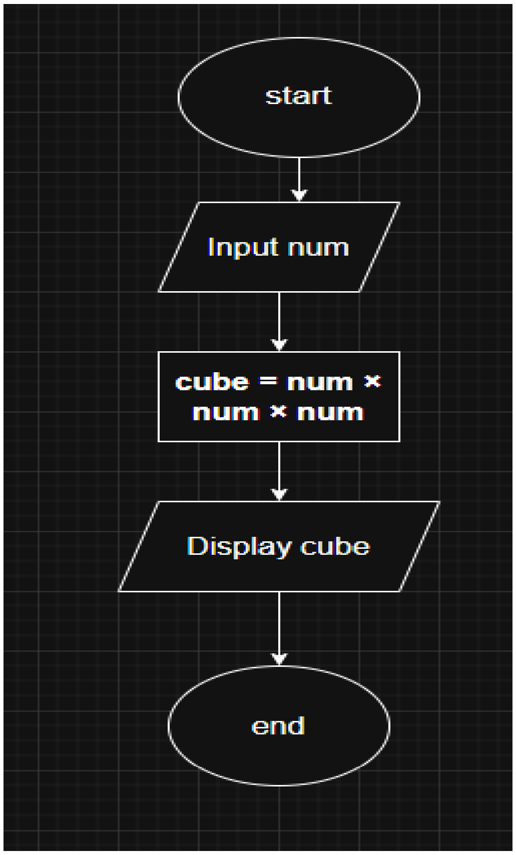
sol:

algorithm:

```
Step 1: Start
Step 2: Input a number → num
Step 3: Calculate cube → cube = num × num × num
Step 4: Display cube
Step 5: End
```

pseudocode:

```
START
  INPUT num
  cube ← num * num * num
  PRINT "Cube of the number is:", cube
END
```

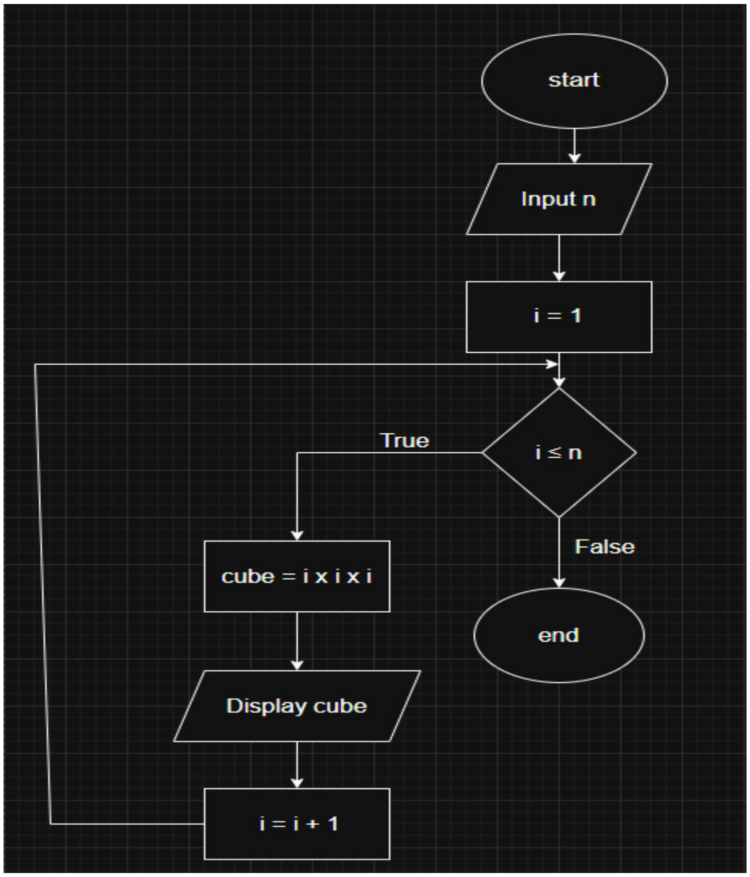


22) To print cubes of first n numbers

sol:

algorithm:

```
Step 1: Start
Step 2: Input n
Step 3: Initialize i ← 1
Step 4: While i ≤ n
            cube ← i × i × i
            Display cube
            i ← i + 1
Step 5: End
```



pseudocode:

```
START
  INPUT n
  i ← 1
  WHILE i ≤ n DO
    cube ← i * i * i
    PRINT "Cube of", i, "is", cube
    i ← i + 1
END
```

23) To find the sum of n given numbers.

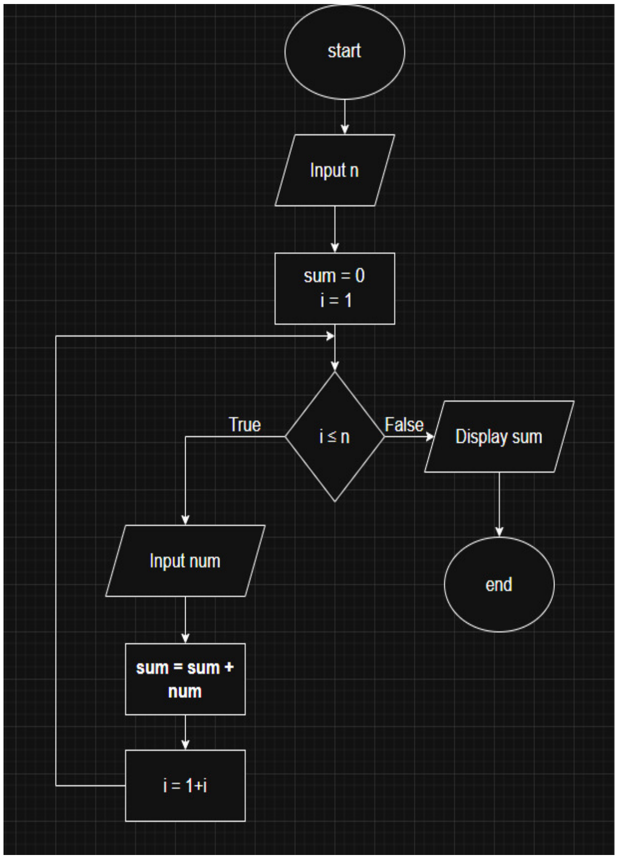
sol:

algorithm:

```
Step 1: Start
Step 2: Input n
Step 3: Initialize sum ← 0
Step 4: For i = 1 to n do
           Input number → num
           sum ← sum + num
Step 5: Display sum
Step 6: End
```

pseudocode:

```
START
  INPUT n
  sum ← 0
  FOR i ← 1 TO n DO
    INPUT num
    sum ← sum + num
  END FOR
  PRINT "Sum of", n, "numbers is:", sum
END
```



24) To find factorial of given number.

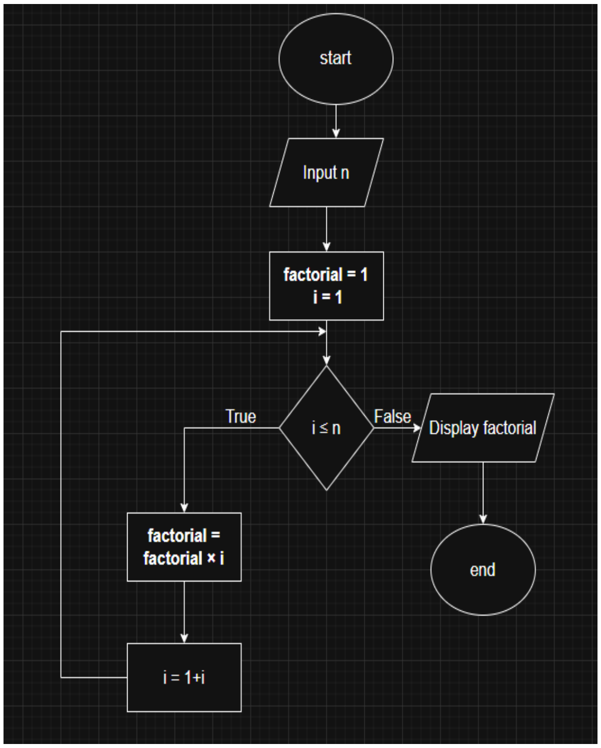
sol:

algorithm:

```
Step 1: Start
Step 2: Input number → n
Step 3: Initialize factorial ← 1
Step 4: For i = 1 to n
             factorial ← factorial × i
Step 5: Display factorial
Step 6: End
```

pseudocode:

```
START
  INPUT n
  factorial ← 1
  FOR i ← 1 TO n DO
    factorial ← factorial * i
  END FOR
  PRINT "Factorial of", n, "is:", factorial
END
```

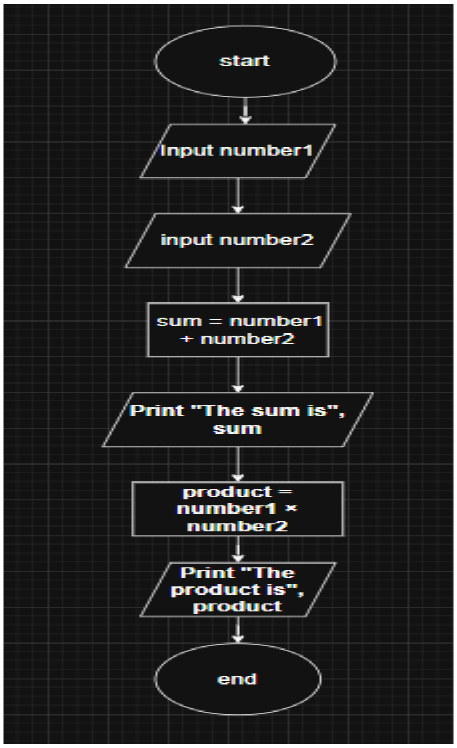


25) Given the following pseudo code:
Use variables sum, product, number1, number2 of type real

```
display "Input two numbers"
accept number1, number2
sum number1 + number2
print "The sum is", sum
product=number1*number2
print "The Product is", product
end program
```

Draw a flow chart for the same and dry run the given pseudocode if it is working fine.

sol:



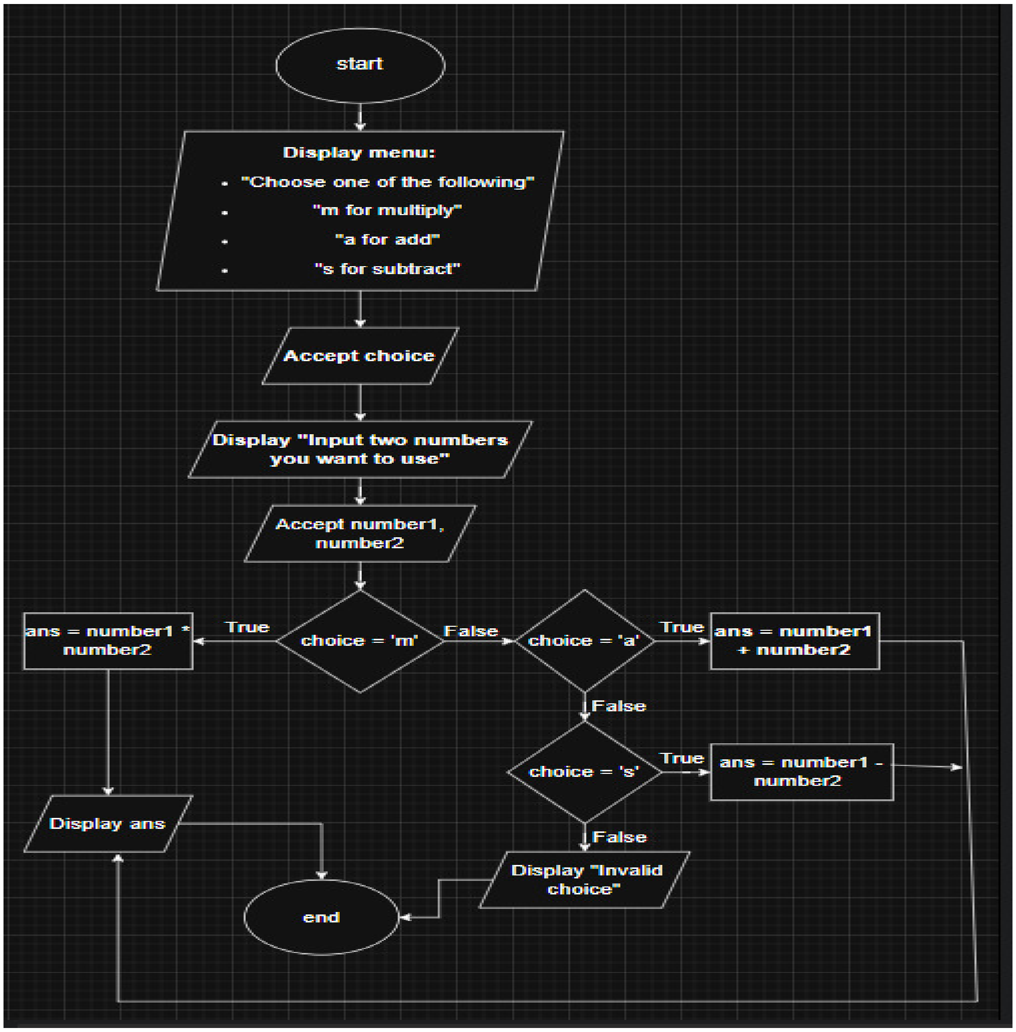
Step	number1	number2	sum	product	Output
1	5	3	–	–	"Input two numbers"
2	5	3	–	–	–
3	5	3	8	–	–
4	5	3	8	15	"The sum is 8"

5	5	3	8	15	"The product is 15"

26) Given the following pseudo code:

```
Use variables: choice, of the type character
ans, number1, number2, of type integer
display "choose one of the following"
display "m for multiply"
display "a for add"
display "s for subtract"
accept choice
display "input two numbers you want to use"
accept number1, number2
if choice = m then ans = number1 * number2
if choice = a then ans = number1 + number2
if choice = s then ans = number1 - number2
display ans
```

Draw a flow chart for the same and dry run the given pseudocode if it is working fine.



sol:

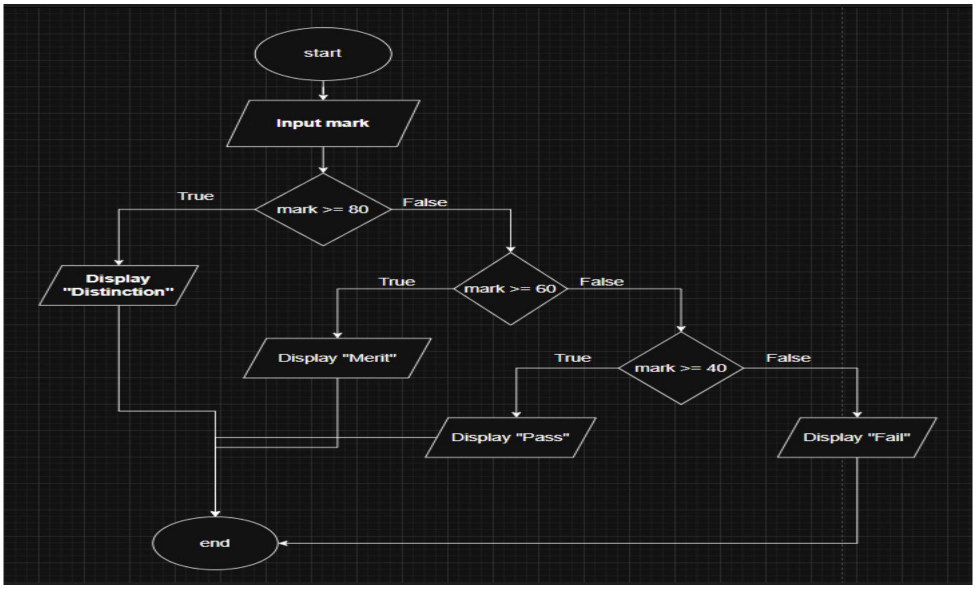
Step	choice	number1	number2	ans	Output
1	'm'	5	3	–	Display menu
2	'm'	5	3	15	Display "Input numbers"
3	'm'	5	3	15	Display ans → 15
4	'a'	8	4	12	Display ans → 12
5	's'	10	6	4	Display ans → 4

27) Given the following pseudo code:

```
Use variables: mark of type integer
If mark >= 80, display "distinction"
If mark >= 60 and mark < 80, display "merit"
If mark >= 40 and mark < 60, display "pass"
If mark < 40 display, "fail"
```

Draw a flow chart for the same and dry run the given pseudocode if it is working fine.

sol:

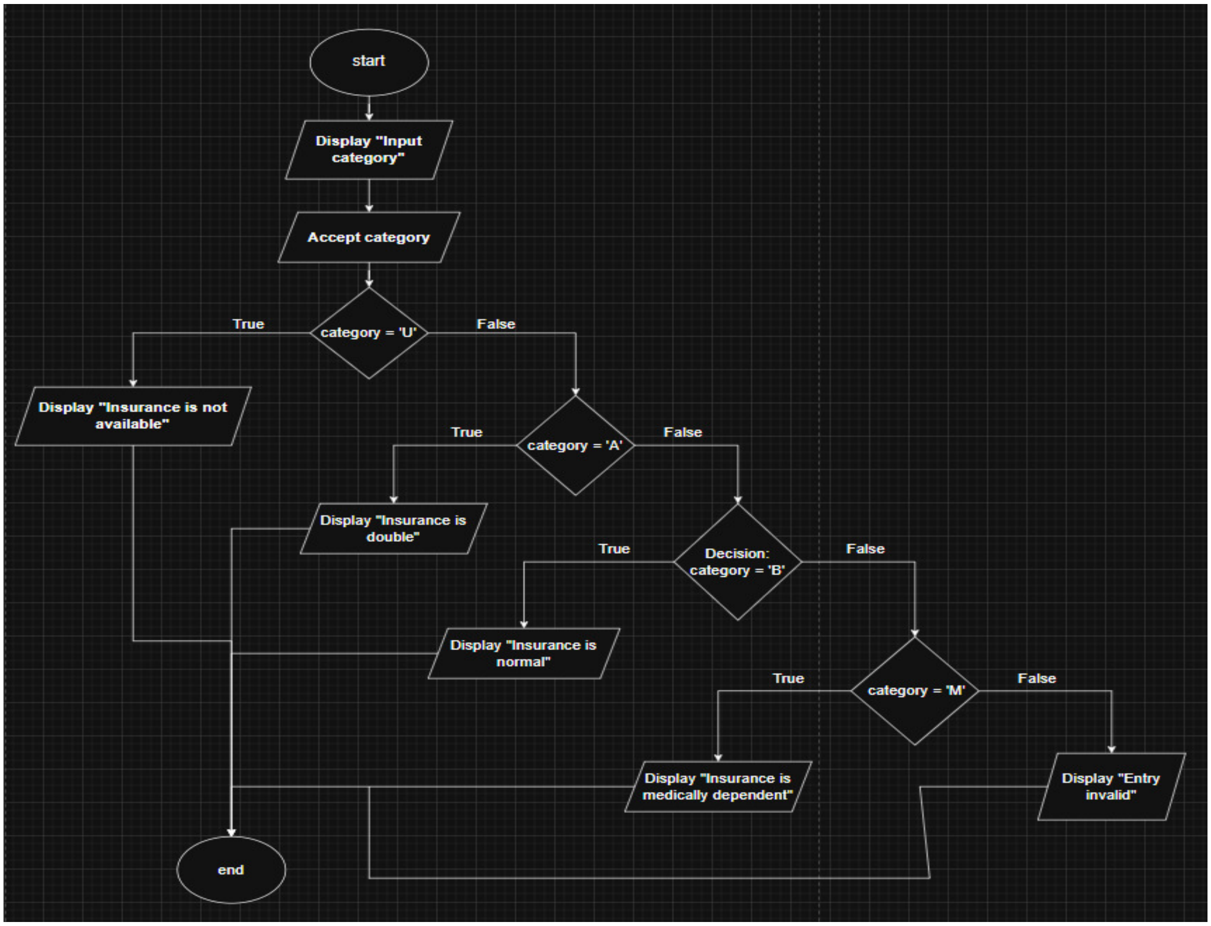


Step	mark	Output
1	85	"Distinction"
2	72	"Merit"
3	55	"Pass"
4	30	"Fail"

28) Given the following pseudo code:

```
Use variables: category of type character
Display "input category"
Accept category
If category = 'U'
   Display "insurance is not available"
Else If category = 'A' then
   Display "insurance is double"
Else If category = 'B' then
   Display "insurance is normal"
Else If category = 'M' then
   Display "insurance is medically dependent"
Else
   Display "entry invalid"
```

Draw a flow chart for the same and dry run the given pseudocode if it is working fine.

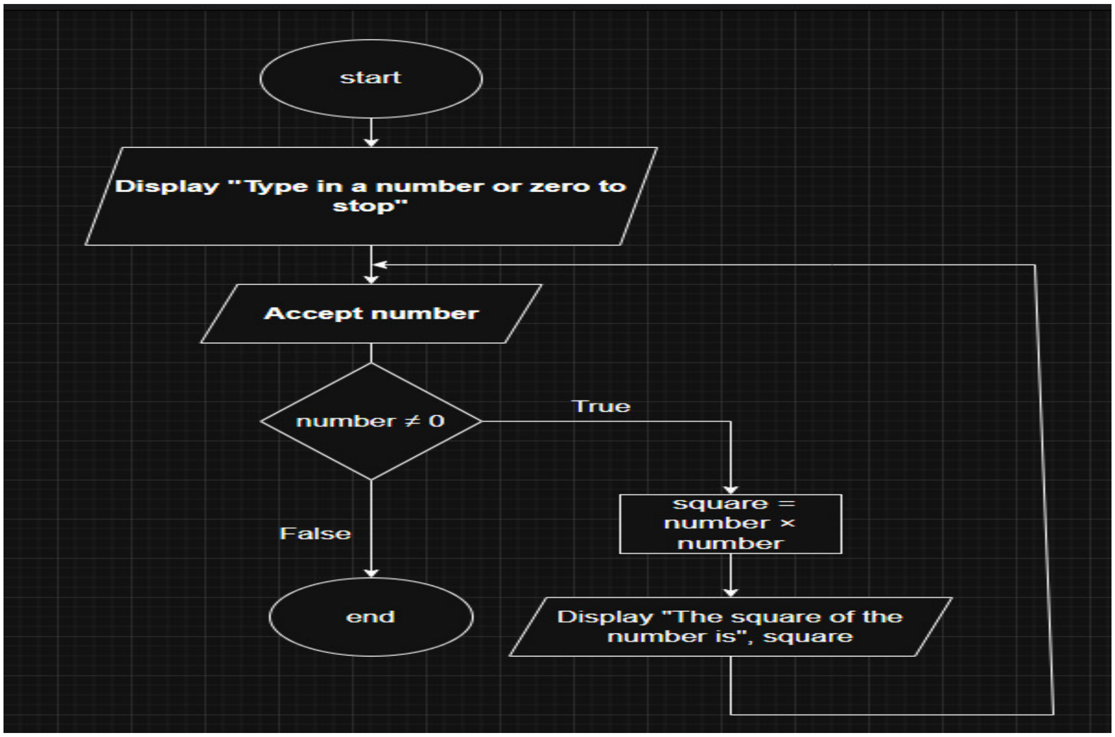


Test Case	category	Output
1	'U'	Insurance is not available
2	'A'	Insurance is double
3	'B'	Insurance is normal
4	'M'	Insurance is medically dependent
5	'X'	Entry invalid

29) Given the following pseudo code :

```
Use variable: number of type real
DISPLAY "Type in a number or zero to stop"
ACCEPT number
WHILE number 0
    Square = number * number
    DISPLAY "The square of the number is", square
    DISPLAY "Type in a number or zero to stop"
    ACCEPT number
ENDWHILE
```

Draw a flow chart for the same and dry run the given pseudocode if it is working fine.


start
Display "Type in a number or zero to stop"
Accept number
number ≠ 0
True
False
square = number × number
Display "The square of the number is", square
end

Step	number	square	Output
1	3	9	"The square of the number is 9"
2	5	25	"The square of the number is 25"
3	0	–	Loop stops, program ends