**PROBLEM SOLVING AND PYTHON PROGRAMMING**

**RECORD**

**FLOWCHART:**

* A flowchart is a graphical representation of the logic for the problem solving.
* The purpose of the flowchart is making the logic of the program in a visual representation.
* Flowcharts is a diagram made up of boxes, diamonds, and other shapes, connected by arrows.
* Each shape represents a step in process and arrows show the order in which they occur.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **NAME OF THE SYMBOL** | **SYMBOL** | **TYPE** | **DESCRIPTION** |
| 1. | Terminal symbol |  | Oval | Represents the start and stop of the program. |
| 2. | Input/Output symbol |  | Parallelogram | Denotes either input or output operation. |
| 3. | Process symbol |  | Rectangle | Denotes the process to be carried out. |
| 4. | Decision symbol |  | Diamond | Represents decision making and branching. |
| 5. | Flow lines |  | Arrows | Represents the sequence of steps and direction of flow. It is used to connect symbols. |
| 6. | Function symbol |  | Subroutine/Function | Represents/calls a pre-defined process. |
| 7. | Connector |  | Circle | A connector symbol is represented by a circle and a letter or digit is placed in the circle to specify the link. This symbol is used to connect flowcharts. |

**TOOLS USED TO DRAW FLOWCHART**

1. **Smart Draw:** A good tool to draw and understand but can’t save the file in system it can be used for free up to 7 days after that we must pay to use it.
2. **Canva:**  A user-friendly tool which allows the user to view in mobile using the application and can be saved in any format. Without even subscription all the features were available.
3. **App.Diagrams.net:**  The diagrams can be saved and also at any destination you want it to be. But the Output Wasn’t precise and not in single page the saved diagrams open up to the website.
4. **Lucidchart:** The diagrams can be directly stored into the system and has all the features and also easy to use. It is required to be paid after some uses .
5. **Visme:** The tool is used for flowchart animation and content creating and in teaching, but more tools are available when you pay for them.
6. **Zenflowchart:**  The diagrams can be directly stored into the system and has all the features and also easy to use. But it restricts to use more than 20 shapes on using the 21st shape it must be paid.
7. **Visual Paradiagram:** Visual paradiagram is explicitly designed for flowchart drawing, it is also paid one to use but in complex algorithm cases it is the best.
8. **Creatly:** This tool is used to design Unified Modelling Language (UML) and flowcharts.
9. **Google Draw:** All the features are available and they are directly stored in the Google Drive. It should be logged in using Email. But the page size was limited, also typing the algorithm wasn’t comfortable.

**DRAW FLOWCHART AND WRITE ALGORITHM AND PSEUDOCODE FOR THE FOLLOWING PROBLEMS**

**EXERCISE NO :** 1(a) **ELECTRICITY BILLING**

**DATE :** 29-11-2022

**AIM :**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1** : Start

**STEP 2** : Get current month and last month units.

**STEP 3** : Enter Old Unit (OU)

**STEP 4** : Calculate N = CU – OU

**STEP 5** : Check the condition N <= 100

**5.1** : If true, Calculate EC using formula FC = 0, DC = 0, EC = 0

**5.2** : Calculate Total charges = FC + DC + EC

**5.3** : Display amount needed to pay and go to Step 9

**STEP 6** : Check for condition N <= 200

**6.1** : If true, Calculate EC using formula FC = 20, DC = 18, EC = (N-100) \* 1.5

**6.2** : Calculate the Total charges = FC + DC + EC

**6.3** : Display amount needed to pay and go to Step 9

**STEP 7** : Check for condition N <= 500

**7.1** : If true, Calculate EC using formula FC = 30, DC = 48, EC =0+(100\*2)+(unit-200)\*3

**7.2** : Calculate the Total charges = FC + DC + EC

**7.3** : Display amount needed to pay and go to Step 9

**STEP 8** : Check for condition N > 500

**8.1** : If true, Calculate EC using formula FC=50, DC=167.2, EC=(100\*3.5)+(300\*4.6)+(unit- 500)\*6.6

**8.2** : Calculate Total charges = FC + DC + EC

**8.3** : Display amount needed to pay and go to Step 9

**STEP 9** : Stop

**PSEUDOCODE :**

START

GET CU

GET OU

CALCULATE N=CU-OU

IF N<=100 THEN FC = 0, DC = 0, EC= 0

CALCULATE EC

ELIF N<=200 THEN FC = 0, DC = 0, EC= 0

CALCULATE EC = (N – 100) \* 1.5

ELIF N<=500 THEN FC = 0, DC = 0, EC= 0

CALCULATE EC = ( N - 100) \* 3.5

ELIF N>500 THEN FC = 0, DC = 0, EC= 0

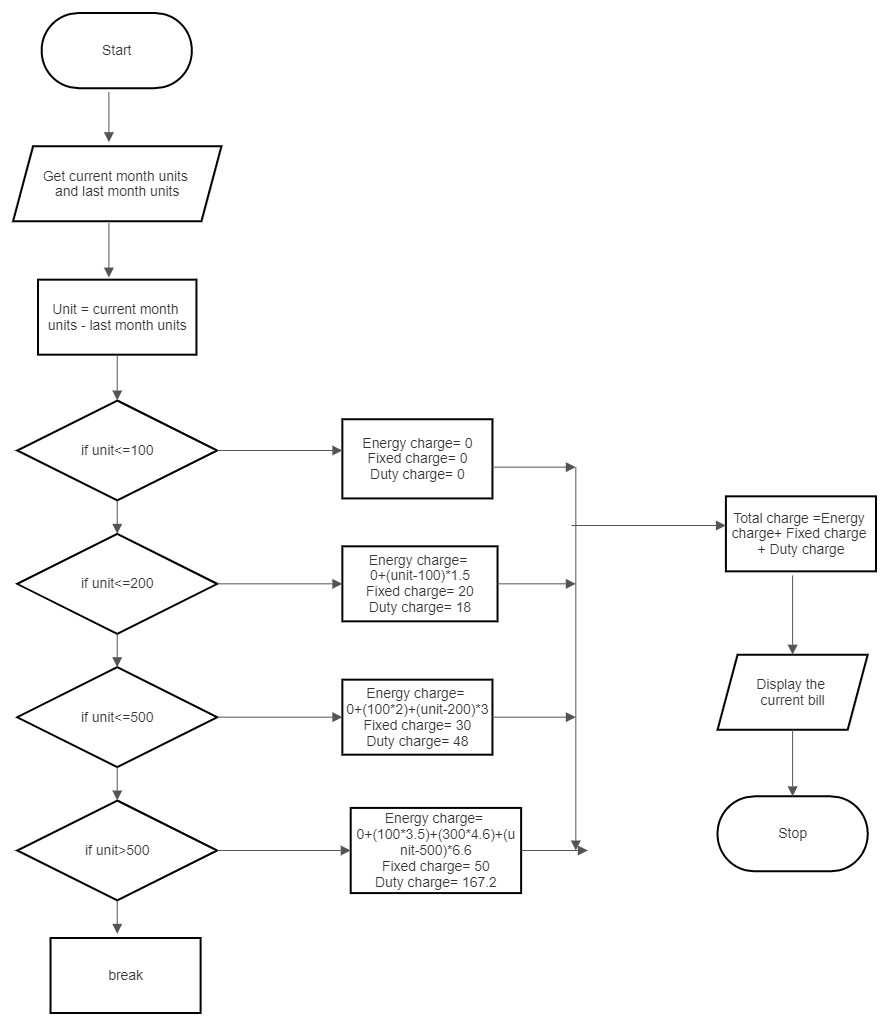
CALCULATE EC = (400 \* 4.5) + (N - 500) \* 6

ENDIF

PRINT Total Charges = FC + DC + EC

STOP

**FLOWCHART :**



**RESULT :**

Thus, the flowchart is drawn and the algorithm and ppseudocode are written for the given problem.

**EXERCISE NO :** 1(b) **RETAIL SHOP BILLING**

**DATE :** 29-11-2022

**AIM :**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Get the Bill number

**STEP 3 :** Get Customer Name, Address and Phone Number

**STEP 4 :** Get the value of total number of items purchased

**STEP 5 :** Initialize the values for i = 0, Total = 0 and Subtotal = 0

**STEP 6 :** Check if condition i <= n

**6.1 :** If true, get Item name, Price, Quantity and Discount

**6.1.1 :** Calculate the Subtotal = Quantity \* Price – Discount

**6.1.2 :** Calculate the Total = Total + Subtotal

**6.1.3 :** Increment the value i and go to Step 6

**6.2 :** If false, get the GST value

**6.2.1 :** Calculate Total\_Bill\_Amount = Total + GST / 100

**6.2.2 :** Display the Total\_Bill\_Amount

**STEP 7 :** Stop

**PSEUDOCODE :**

START

GET Bill Number

GET Customer name , Address and Phone number

GET Number of items purchased as n

INITIALIZE i=0, total=0, subtotal=0

IF I<=n

GET Item Name, Price, Count, Discount

CALCULATE subtotal = Price \* Count-Discount/100

CALCULATE total=total+subtotal

i=i+1

ELSE

GET GST value

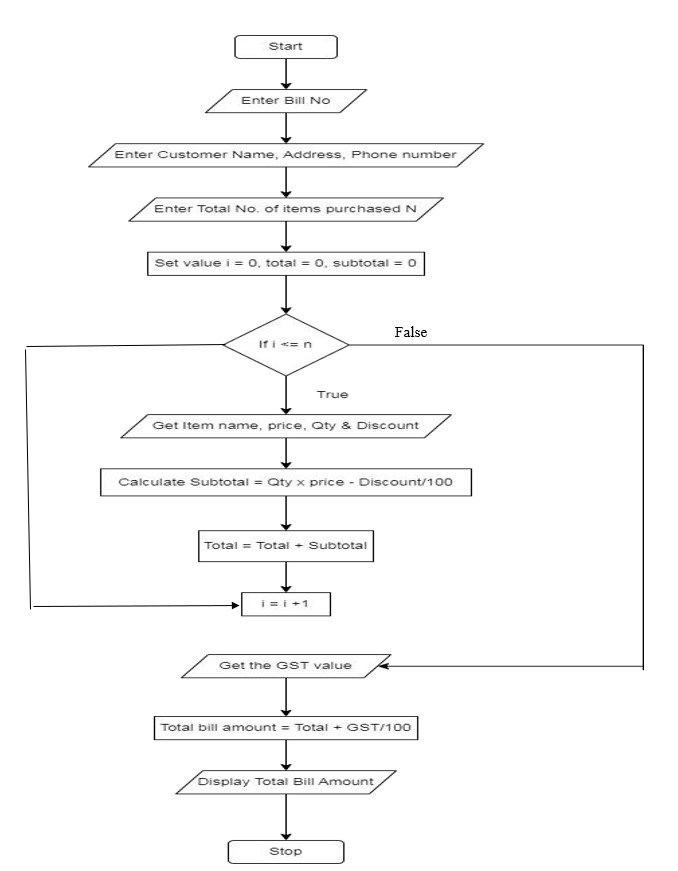
CALCULATE the Total\_bill\_amount = Total + GST value

PRINT Total\_bill\_amount

ENDIF

STOP

**FLOWCHART :**



**RESULT :**

Thus, the flowchart is drawn and the algorithm and pseudocode are written for the given problem.

**EXERCISE NO :** 1(c) **SINE SERIES**

**DATE :** 29-11-2022

**AIM :**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Get the value of X

**STEP 3 :** Initialize the values of i = 1, sine = 0 and import math

**STEP 4 :** Get the value of N

**STEP 5 :** Check whether the value of i is less than N

**5.1 :** If condition is true, convent x to radian and add it to y

**5.1.1 :** Let value of s be (-1) to the power i

**5.1.2 :** Now calculate the series using the formula sine=sine+((y\*\*2\*i+1))/math.factorial(2\*i + 1))\* s

**5.1.3 :** Increment value of i by 1

**5.2 :** If condition is false display sine

**STEP 6 :** Stop

**PSEUDOCODE :**

START

GET the value of X

INITIALIZE i=1, sine=0, AND IMPORT math

GET the value of N

IF i<N THEN

Y=Y+X\*(3.1416/180)

S=(-1)\*\*1

sine=sine+((Y\*\*2\*i+1)/math.factorial(2\*i+1))\*s

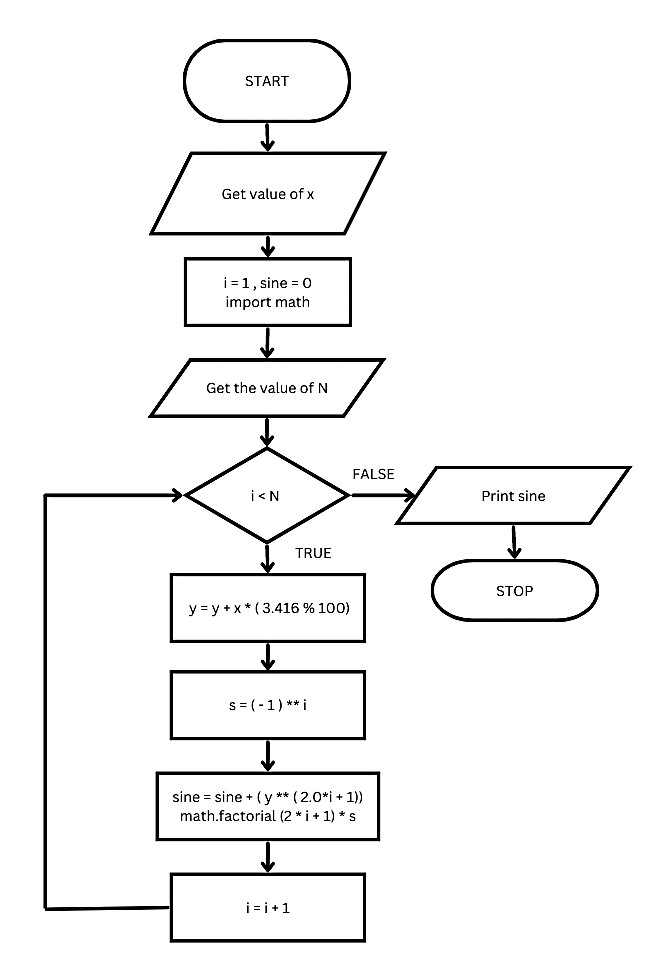
i=i+1

ELSE

PRINT sine

STOP

**FLOWCHART :**



**RESULT :**

Thus, the flowchart is drawn and the algorithm and pseudocode are written for the given problem.

**EXERCISE NO :** 1(d) **WEIGHT OF THE MOTORBIKE**

**DATE :** 29-11-2022

**AIM :**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1** : Start.

**STEP 2** : Get Gross Vehicle Weight Rating (GVWR)

**STEP 3** : Get Dry weight (DW)

**STEP 4** : Get Fuel weight (FW)

**STEP 5** : Get Raider weight (RW)

**STEP 6** : Get Passenger weight(PW)

**STEP 7** : Calculate Total weight = DW+FW+RW+PW

**STEP 8** : Get Load.

**STEP 9** : Calculate Load\_weight=Total weight + Load

**STEP 10** : Calculate safe weight. GVWR-Load-weight.

**STEP 11** : Check the condition safe weight >=0.

* 1. : If true, print the message “You have a safe load and you can drive" GOTO step 12
  2. : If false, print the message "Reduce the load and then drive”.

GOTO step 8.

**STEP 12** : Stop.

**PSEUDOCODE :**

START

GET GVWR

GET DW

GET FW

GET RW

GET PW

CALCULATE Total Weight = DW + FW+ RW + PW

GET Load

CALCULATE Load Weight = Total Weight + Load

CALCULATE Safe\_Weight = GVWR - Load Weight

IF Safe\_Weight >= 0 THEN

PRINT You have a safe load and you can drive

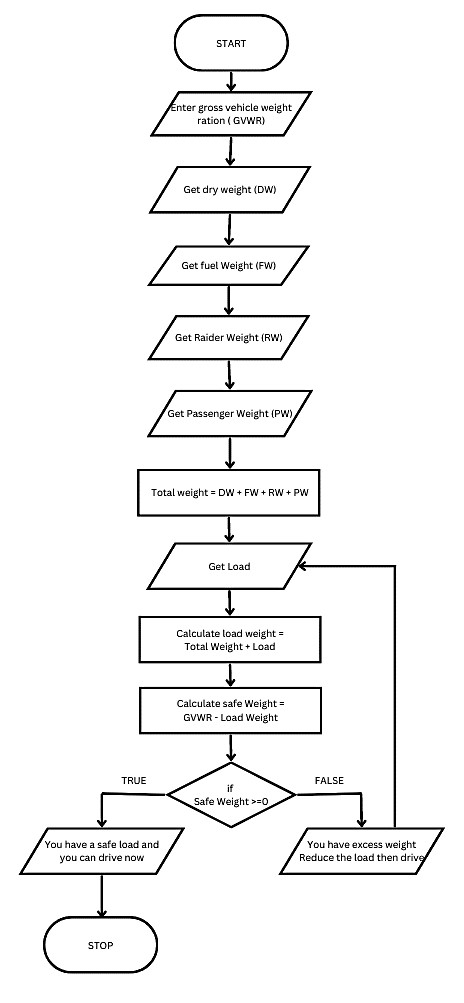
ELSE

PRINT You have excess weight, Reduce the load and then drive

ENDIF

STOP

**FLOWCHART :**



**RESULT :**

Thus, the flowchart is drawn and the algorithm and pseudocode are written for the given problem.

**EXERCISE NO :** 1(e) **WEIGHT OF A STEEL BAR**

**DATE :** 29-11-2022

**AIM :**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Get the number of iron rod required (n)

**STEP 3 :** Initialize i = 0 and total weight = 0

**STEP 4 :** Check if the value of i is less than n

**4.1 :** If true, get the diameter of the rod (D)

**4.1.1 :** Calculate the unit weight using formula D\*\*2 / 162 = W

**4.1.2 :** Get the number of rod with diameter D

**4.1.3 :** Calculate the weight of the rod using formula Number of Rod \* D \* Unit Weight

**4.1.4 :** Add the weight to Total

**4.1.5 :** Increment the value of i by 1

**4.2 :** If condition is false, Display total as total weight of the rod

**STEP 5 :** Stop

**PSEUDOCODE :**

START

GET n

INITIALIZE i=1,total\_weight=0

IF i <= n THEN

GET D

CALCULATE weight = D\*2 /162

CALCULATE total\_weight = total\_weight + weight

i=i+1

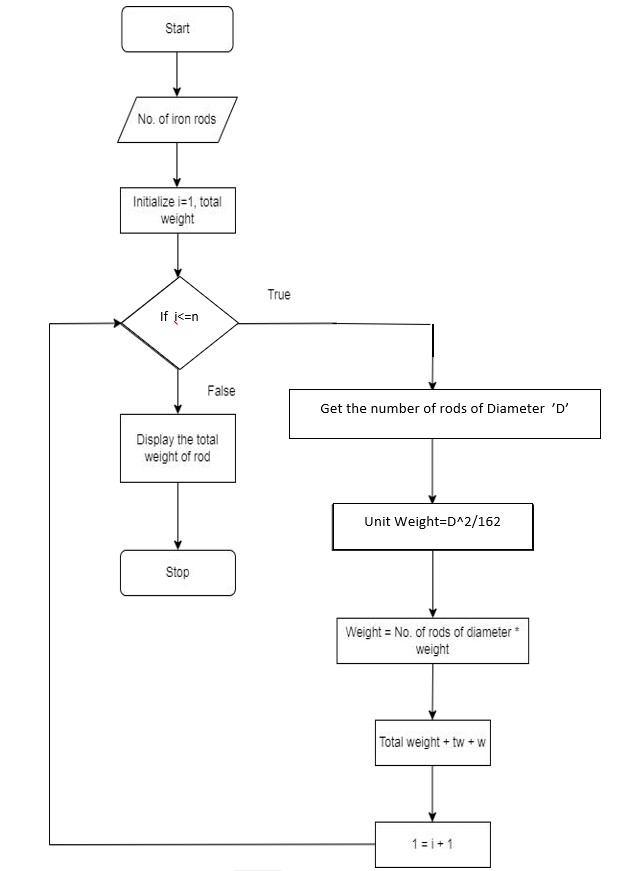
ELSE

PRINT total\_weight

ENDIF

STOP

**FLOWCHART :**



**RESULT:**

Thus, the flowchart is drawn and the algorithm and pseudocode are written for the given problem.

**EXERCISE NO :** 1(f) **STUDENT GRADE ANALYSIS**

**DATE :** 29-11-2022

**AIM :**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Get the number of students (N)

**STEP 3 :** Assign i = 0

**STEP 4 :** Check for the condition i < N

**4.1 :** If true, Get Name, Roll Number and Marks m1, m2, m3, m4, m5

**4.2 :** Calculate Total = m1 + m2 + m3 + m4 + m5 and Average = Total / 5

**4.3 :** Display Name and Roll Number

**4.4 :** Check for condition avg >= 30 and avg < 50

**4.4.1 :** If true, Display the message "Your grade is C" and increase i value by 1

**4.5 :** Check for condition avg > 50 and avg < 80

**4.5.1 :** If true, Display the message "Your grade is B" and increase i value by 1

**4.6 :** Check or the condition avg > 80 and avg ≤ 100

**4.6.1 :** If true, Display the message "Your grade is A" and increase i value by 1

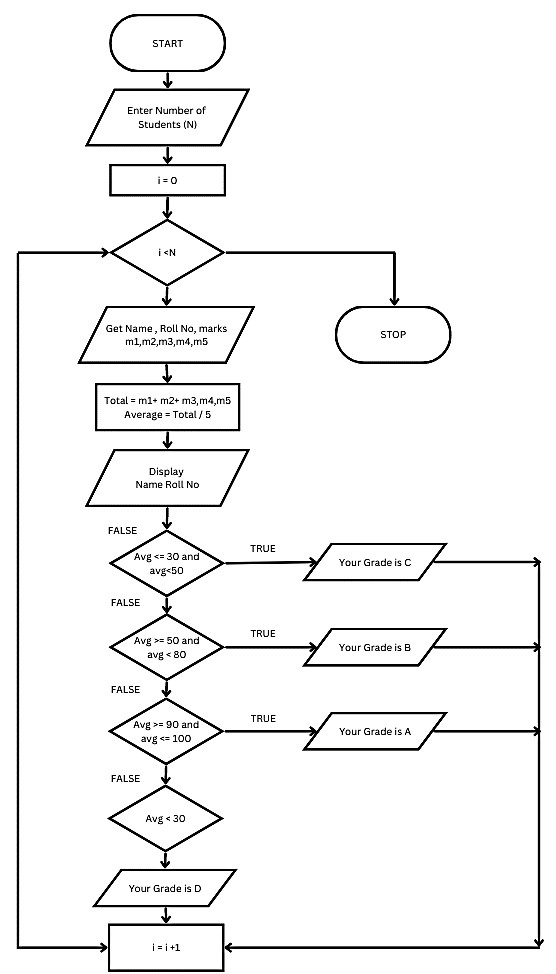
* 1. **:** Check for the condition avg < 30

**4.7.1 :** If true, Display the message "Your grade is D"

**STEP 5 :** If false, go to step 6

**STEP 6 :** Stop

**FLOWCHART :**



**RESULT :**

Thus, the flowchart is drawn and the algorithm and pseudocode are written for the given problem.

**EXERCISE NO :** 1(g) **COMPUTE THE ELECTRICAL CURRENT IN 3 PHASE AC CIRCUIT**

**DATE :** 29-11-2022

**AIM:**

To draw flowchart and write algorithm and pseudocode for the given problem.

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Get value of Power Factor (PF)

**STEP 3 :** Get value of Current (I)

**STEP 4 :** Get value of voltage (V)

**STEP 5 :** Calculate P using the formula P = √3 \* PF \* I \* V

**STEP 6 :** Display the value of P

**STEP 7 :** Stop

**PSEUDOCODE :**

START

GET PF

GET I

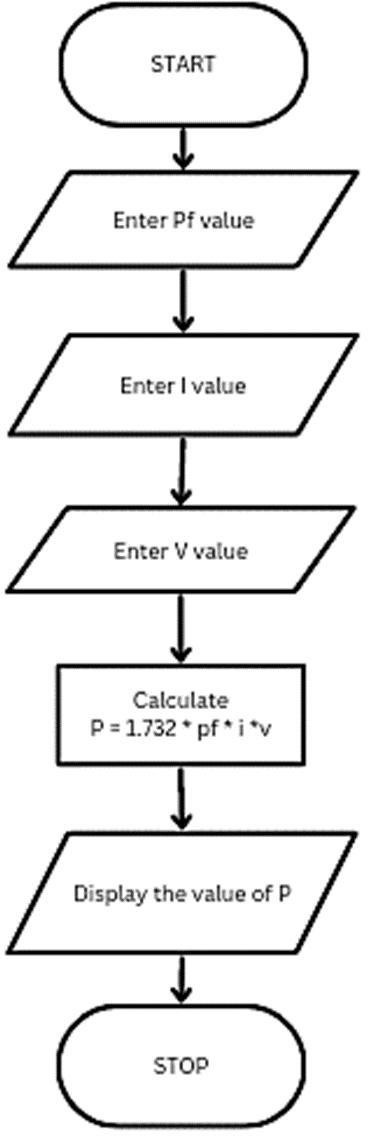
GET V

CALCULATE P = √3 \* PF \* I \* V

PRINT P

STOP

**FLOWCHART :**



**RESULT :**

Thus, the flowchart is drawn and the algorithm and pseudocode are written for the given problem.