Expt. No. : 1(a) **ELECTRICITY BILL GENERATION**

Date : 21-11-2022

Aim:

To draw flowchart and algorithm for calculating the electricity bill.

Algorithm:

Step 1 : Start

Step 2 : read last month unit and present month unit

Step 3 : unit=present month – last month units

Step 4 : Check if unit<=100

Step 4.1 : If Yes, then Amount=0 and go to Step 5

Step 4.2 : If No, then check if unit<=200

Step 4.2.1 : If Yes, then assign DC=1.8, FC=20, Amount = (unit-100)*1.5

And go to Step 5

Step 4.2.2 : If No, then check if units<=500

Step 4.2.2.1 : If Yes, then Amount = (units-100)*3.5, EC=3.5, DC=48, FC=30

And go to Step 5

Step 4.2.2.2 : If No, then Amount = (unit-500)*6 + (400*4.5). DC=100, FC=75

And go to Step 5

Step 5 : Bill amount = Amount + DC + FC

Step 6 : display Bill amount

Step 7 : Stop

22CSEA49 -NITHISH GOWTHAM.R

Flowchart:

Start

read last_unit,present_unit

unit=present_unit-last_unit

if unit<=100 — Yes → EC=0 DC=0 FC=0 → Amount=0

No

if unit<=200 — Yes → EC=1.5 DC=18 FC=20 → Amount=(unit-100)*1.5

No

if unit<=500 — Yes → EC=3.5 DC=48 FC=30 → Amount=(unit-100)*3.5

No

Amount=(unit-500)*6+(400*4.5) DC=100 FC=75

BIll_amount=Amount+DC+FC

Stop
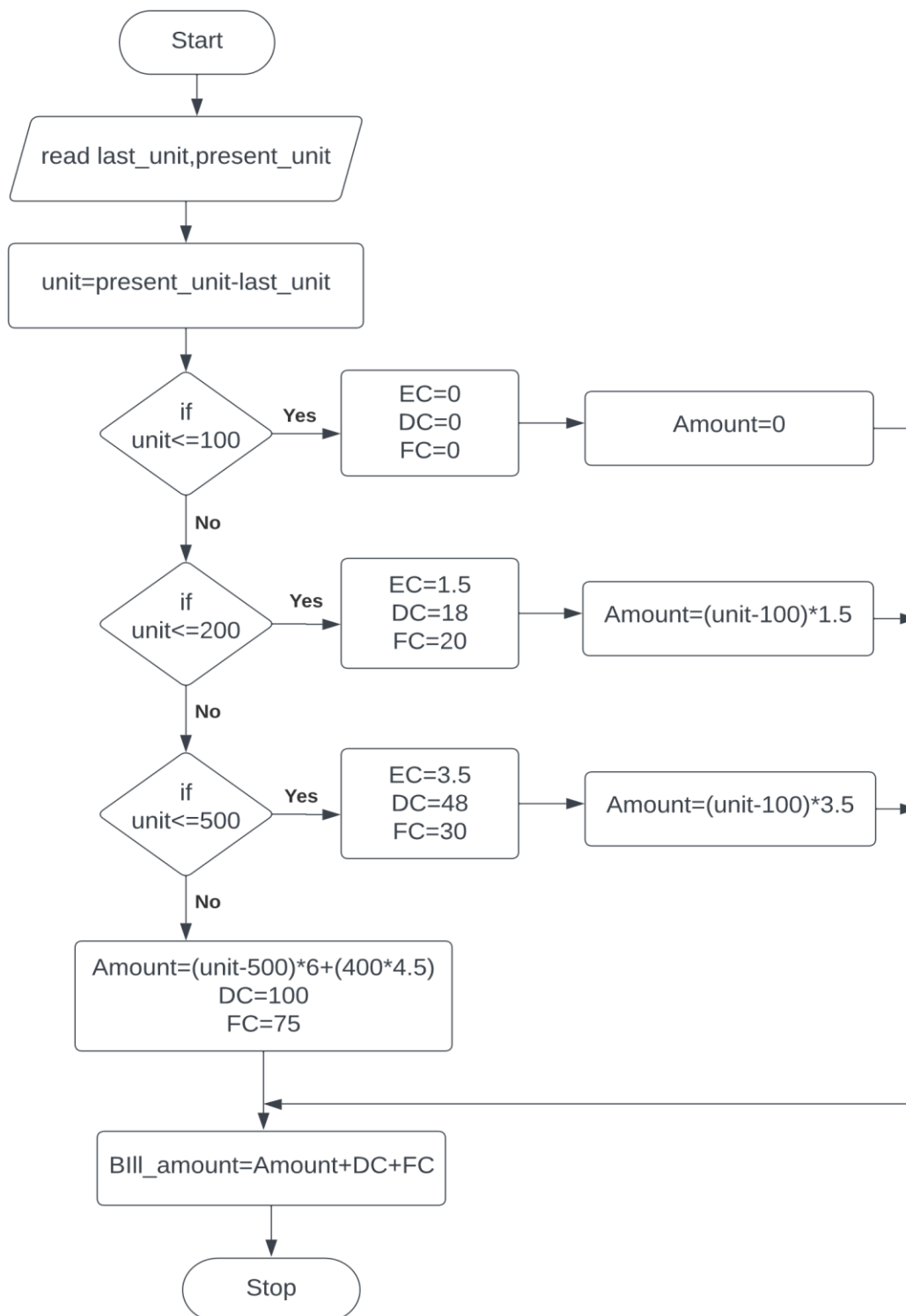
Pseudocode:

BEGIN

GET last_unit, present_unit

COMPUTE = present_unit – last_unit

IF unit<=100

    ASSIGN EC=0, DC=0, FC=0, amount=0

    ELSE IF unit<=200

        ASSIGN EC=1.5, DC=18, FC=20

        COMPUTE amount=(unit-100)*1.5

          ELSE IF unit<=500

            ASSIGN EC=3.5, DC=48, FC=30

            COMPUTE amount=(unit-100)*3.5

        ELSE

          ASSIGN DC=100, FC=75

          COMPUTE (unit-500)*6+(400*4.5)

END IF

COMPUTE bill amount=amount+DC+FC

PRINT bill amount

STOP

Result:

    The algorithm and flowchart is written for the given problem
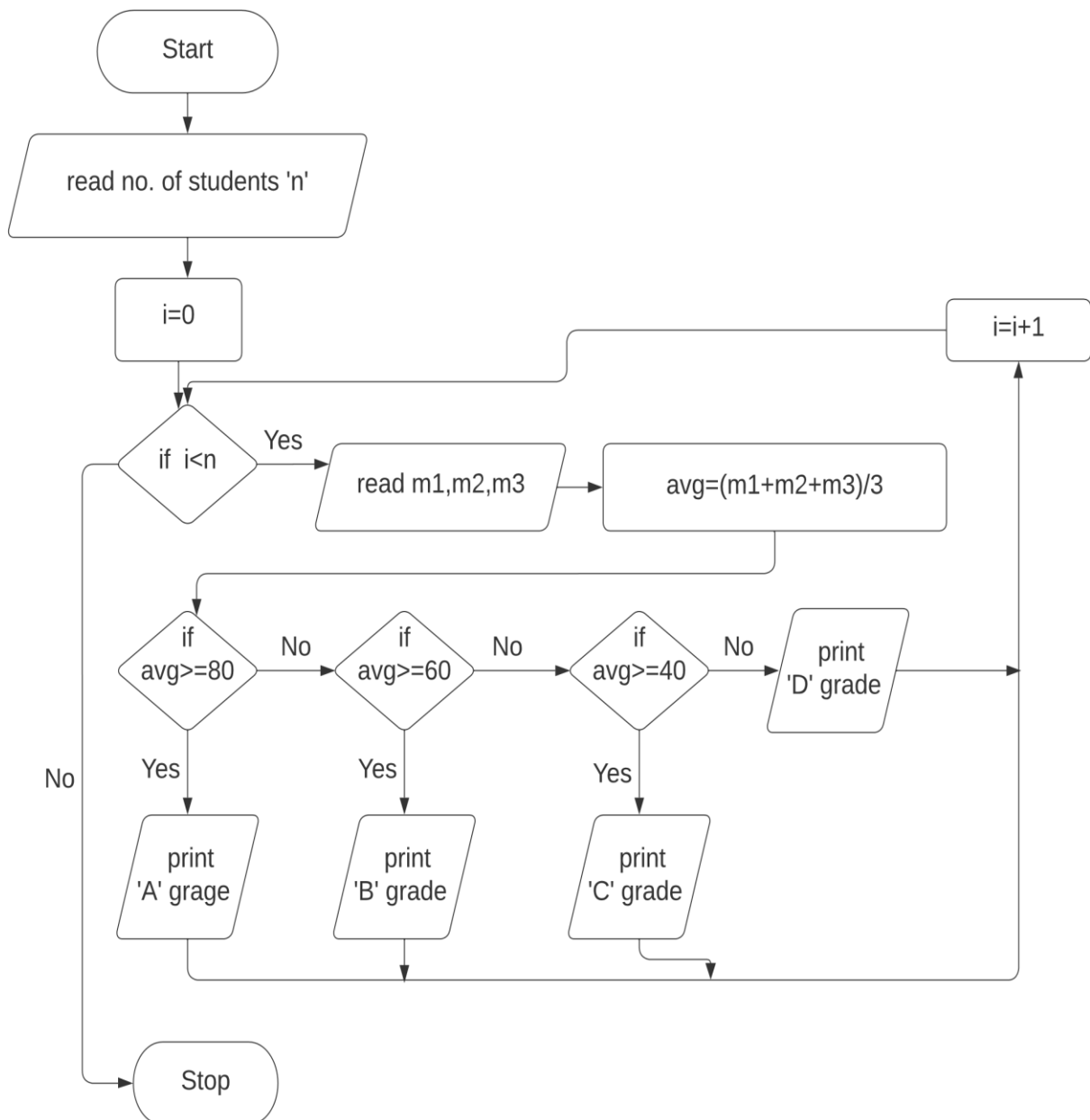
Aim:

To draw flow chart and write algorithm for Student Mark Analysis.

Algorithm:

Step 1	**:**	Start
Step 2	**:**	read No. of students as 'n'
Step 3	**:**	initialize the counter i=0
Step 4	**:**	check if i<n
Step 4.1	**:**	If yes, read m1, m2, m3
Step 4.2	**:**	Compute avg= (m1+m2+m3)/3
Step 4.3	**:**	Check if avg>=80
Step 4.3.1	**:**	If Yes, print 'A-grade' and i=i+1 and go to step 4
Step 4.3.2	**:**	If No, check if avg>=60
Step 4.3.2.1	**:**	If Yes, print 'B-grade' and i=i+1 and go to step 4
Step 4.3.2.2	**:**	If No, check if avg>=40
Step 4.3.2.2.1	**:**	If Yes, print 'C-grade' and i=i+1 and go to step 4
Step 4.3.2.2.2	**:**	If No, print 'D-grade' and i=i+1 and go to step 4
Step 4.4	**:**	If No, then go to step 5
Step 5	**:**	Stop

Flowchart:

```
                    Start
                      |
                      v
        read no. of students 'n'
                      |
                      v
                    i=0                                              i=i+1
                      |                                                ^
                      v                                                |
                  if i<n  ──Yes──>  read m1,m2,m3  ──>  avg=(m1+m2+m3)/3
                      |                                                |
                      No                                               |
                      |              v
                      |          if                if              if          No     print
                      |        avg>=80  ──No──>  avg>=60  ──No──>  avg>=40  ──No──>  'D' grade
                      |           |                 |                 |
                      |          Yes               Yes               Yes
                      |           |                 |                 |
                      |         print             print             print
                      |        'A' grage          'B' grade         'C' grade
                      |           |                 |                 |
                      v
                    Stop
```

Pseudocode:

BEGIN

GET n   // number of students

ASSIGN i=0

WHILE I<N

    GET m1, m2, m3

    COMPUTE avg= (m1+m2+m3)/3

    IF avg>=80

        PRINT 'A-grade'

        I=i+1

    ELSE IF avg>=60

        PRINT 'B-grade'

        I=i+1

    ELSE IF avg>=40

        PRINT 'C-grade'

        I=I+1

    ELSE

        PRINT 'D-grade'

        I=I+1

    END IF

END WHILE

END

Result:

    The algorithm and flowchart is written for the given problem

Aim:

To draw flowchart and algorithm for the given problem

Algorithm:

Step 1   : Start

Step 2   : read value of No. of rods as 'n'

Step 3   : initialise the counter as i=0

Step 4   : Total_weight=0

Step 5   : Check if i<n

Step 5.1   : If Yes, then read Diameter, length of rod
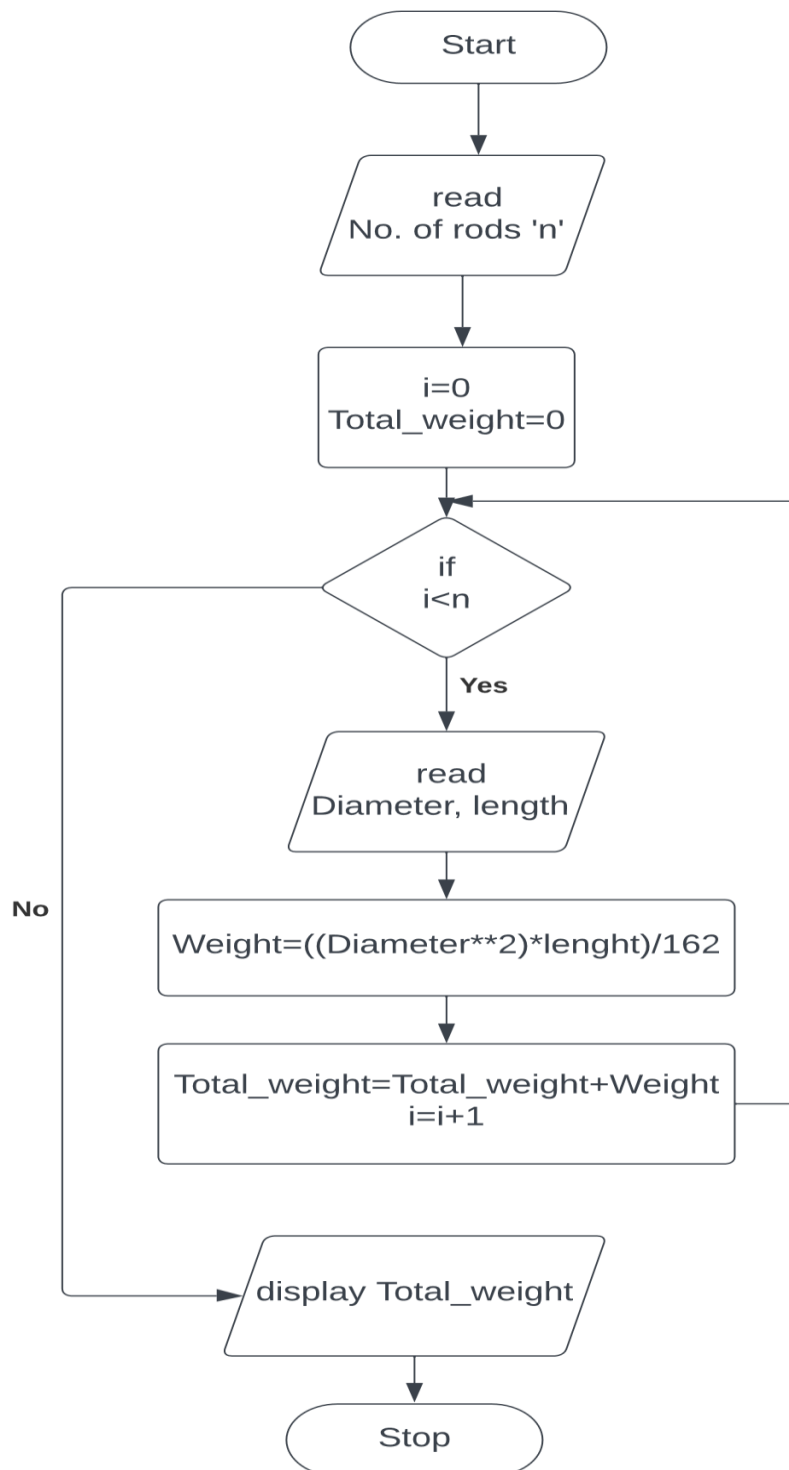
Step 5.2   : Compute Weight=((Diameter**D)*length)/162

Step 5.3   : Compute Total_weight=Total_weght + Weight , i=i+1 and go to step 5

Step 5.4   : If No, then go to step 6

Step 6   : display Total_weight

Step 7   : Stop

Flowchart:

Start

read
No. of rods 'n'

i=0
Total_weight=0

if
i<n

Yes

read
Diameter, length

Weight=((Diameter**2)*lenght)/162

Total_weight=Total_weight+Weight
i=i+1

No

display Total_weight

Stop

Pseudocode:

BEGIN

GET n   // number of rods

ASSIGN I=0, Total weight=0

IF i<n

      GET Diameter, length

      COMPUTE weight=((Diameter**2)*length)/162

      COMPUTE Total weight=Total weight+weight

      COMPUTE i=i+1

END IF

PRINT Total weight

END

Result:

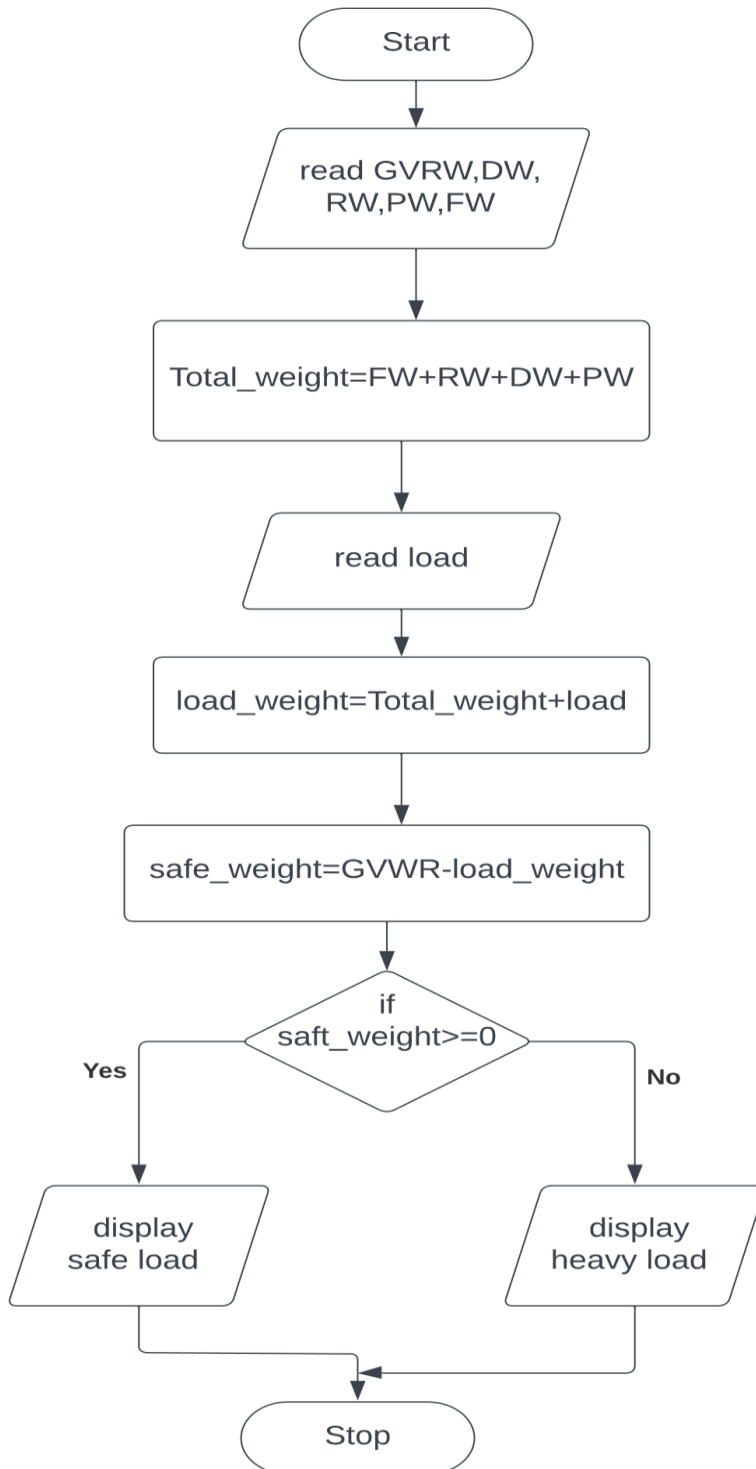   The algorithm and flowchart is written for the given problem.

Aim:

To draw flowchart and write algorithm for the given problem.

Algorithm:

Step 1   : Start

Step 2   : read GVRW, DW, RW, PW, FW

Step 3   : Compute Total weight= FW+RW+DW+PW

Step 4   : read the value of load

Step 5   : Compute load weight = Total weight+load

Step 6   : Compute safe weight = GVWR – load weight

Step 7   : Check if safe weight >=0

Step 7.1   : If Yes, then display safe load and go to step 8

Step 7.2   : If No, then display heavy load and go to step 8

Step 8   : Stop

Flowchart:

Start

read GVRW,DW,
RW,PW,FW

Total_weight=FW+RW+DW+PW

read load

load_weight=Total_weight+load

safe_weight=GVWR-load_weight

if
saft_weight>=0

Yes

No

display
safe load

display
heavy load

Stop

Pseudocode:

BEGIN

GET GVRW, DW, RW, PW, FW

COMPUTE Total weight- FW+RW+DW+PW

GET load

COMPUTE load weight = Total weight+load

COMPUTE safe weitht = GVWR – load weight

IF safe weight>=0

      PRINT safe load

ELSE

      PRINT heavy load

END IF

END

Result:

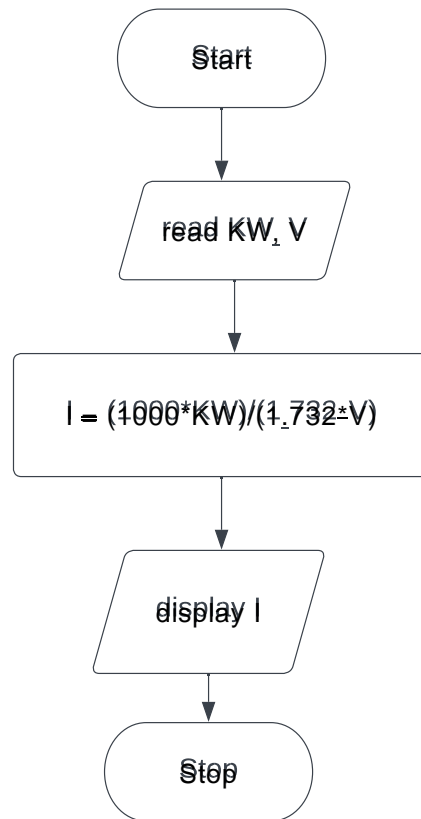   The algorithm and flowchart written for the given problem.

Aim:

To draw flowchart and write algorithm for the given problem

Algorithm:

Step 1   : Start

Step 2   : read the values of KW and V

Step 3   : Compute I=(1000*KW)/(1.732*V)

Step 4   : display I

Step 5   : Stop

Flowchart:

Start

read KW, V

$I = (1000*KW)/(1.732*V)$

display I

Stop

Pseudocode:

BEGIN

GET KW, V

COMPUTE I = (1000*KW)/(1.732*V)

PRINT I

END

Result:

   The algorithm and flowchart is written for the given problem.

Aim:

To draw flowchart and algorithm for retail shop billing.

Algorithm:

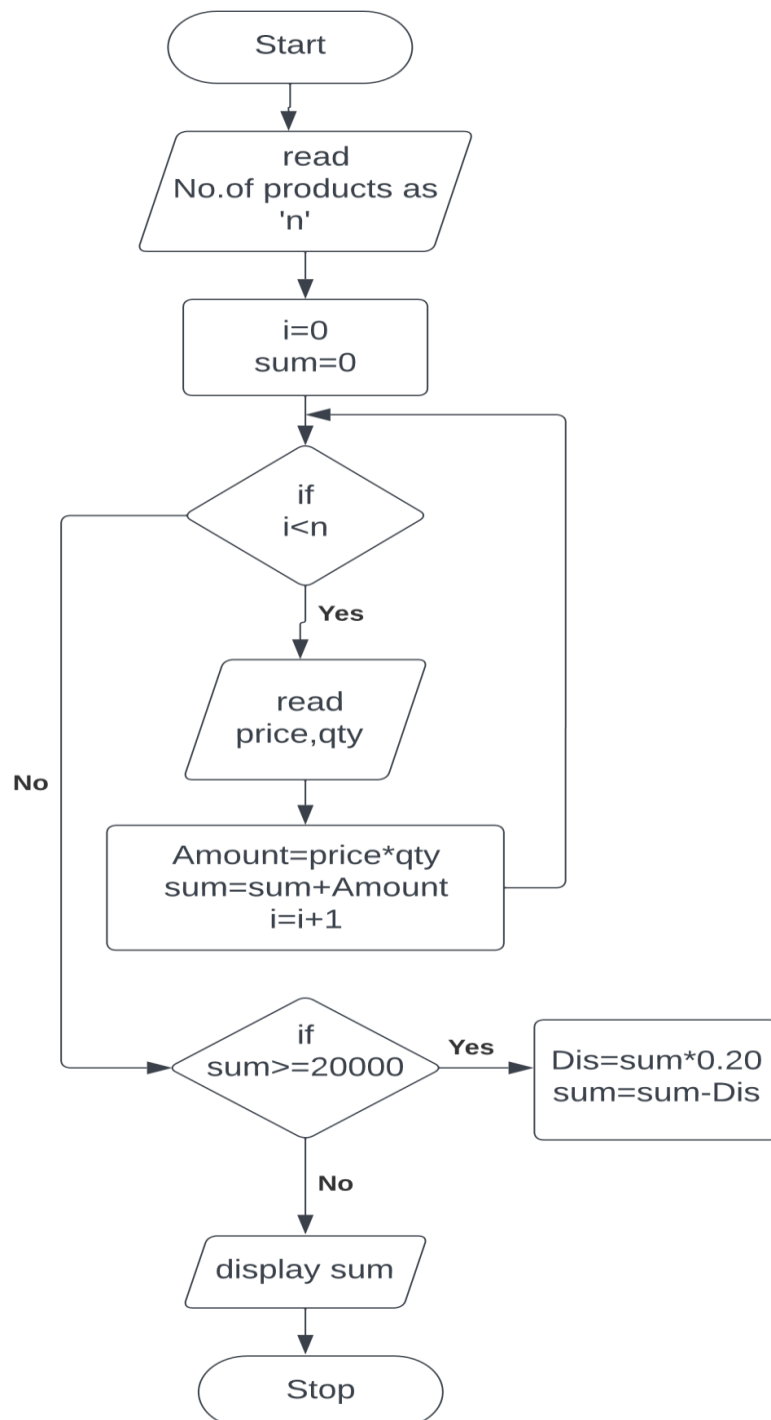|		
Step 1	:	Start
Step 2	:	read no of products 'n'
Step 3	:	initialise the counter i=0
Step 4	:	sum=0
Step 5	:	Check if i<n
Step 5.1	:	If Yes, then read prince, quantity
Step 5.2	:	Compute amount=price*qty, sum=sum+amount, i=i+1
		And go to step 5
Step 5.3	:	If No, go to stpe 6
Step 6	:	Check if sum>=2000
Step 6.1	:	If Yes, then Dis=sum*0.20, sum=sum-Dis
Step 6.2	:	If No, then display sum and go to step 7
Step 7	:	Stop

Flowchart:

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
                    ╱─────────────╲
                   ╱     read       ╲
                  │ No.of products as │
                   ╲      'n'        ╱
                    ╲───────────────╱
                           │
                           ▼
                    ┌─────────────┐
                    │    i=0      │
                    │   sum=0     │
                    └─────────────┘
                           │
                           ▼
                        ╱─────╲
           ┌───No─────╱   if    ╲
           │          ╲   i<n   ╱
           │           ╲──────╱
           │              │ Yes
           │              ▼
           │         ╱─────────╲
           │        ╱   read     ╲
           │        │  price,qty  │
           │         ╲───────────╱
           │              │
           │              ▼
           │      ┌──────────────────┐
           │      │ Amount=price*qty │
           │      │ sum=sum+Amount   │
           │      │    i=i+1         │
           │      └──────────────────┘
           │
           ▼
        ╱─────────╲
       ╱    if      ╲  Yes    ┌──────────────┐
      │ sum>=20000   │───────▶│ Dis=sum*0.20 │
       ╲────────────╱         │ sum=sum-Dis  │
           │                  └──────────────┘
           │ No
           ▼
     ╱──────────────╲
    │  display sum   │
     ╲──────────────╱
           │
           ▼
    ┌─────────────┐
    │    Stop     │
    └─────────────┘
```

Pseudocode:

BEGIN

GET n   // number of products

ASSIGN I=0, sum=0

WHILE i<n

       GET price, qty

       COMPUTE amount=price*qty

       COMPUTE sum=sum+amount

       COMPUTE i=i+1

END WHILE

IF sum>=2000

       COMPUTE Dis=sum*0.20

       COMPUTE sum=sum-Dis

ELSE

       CONTINUE

END IF

PRINT sum

END

Result:

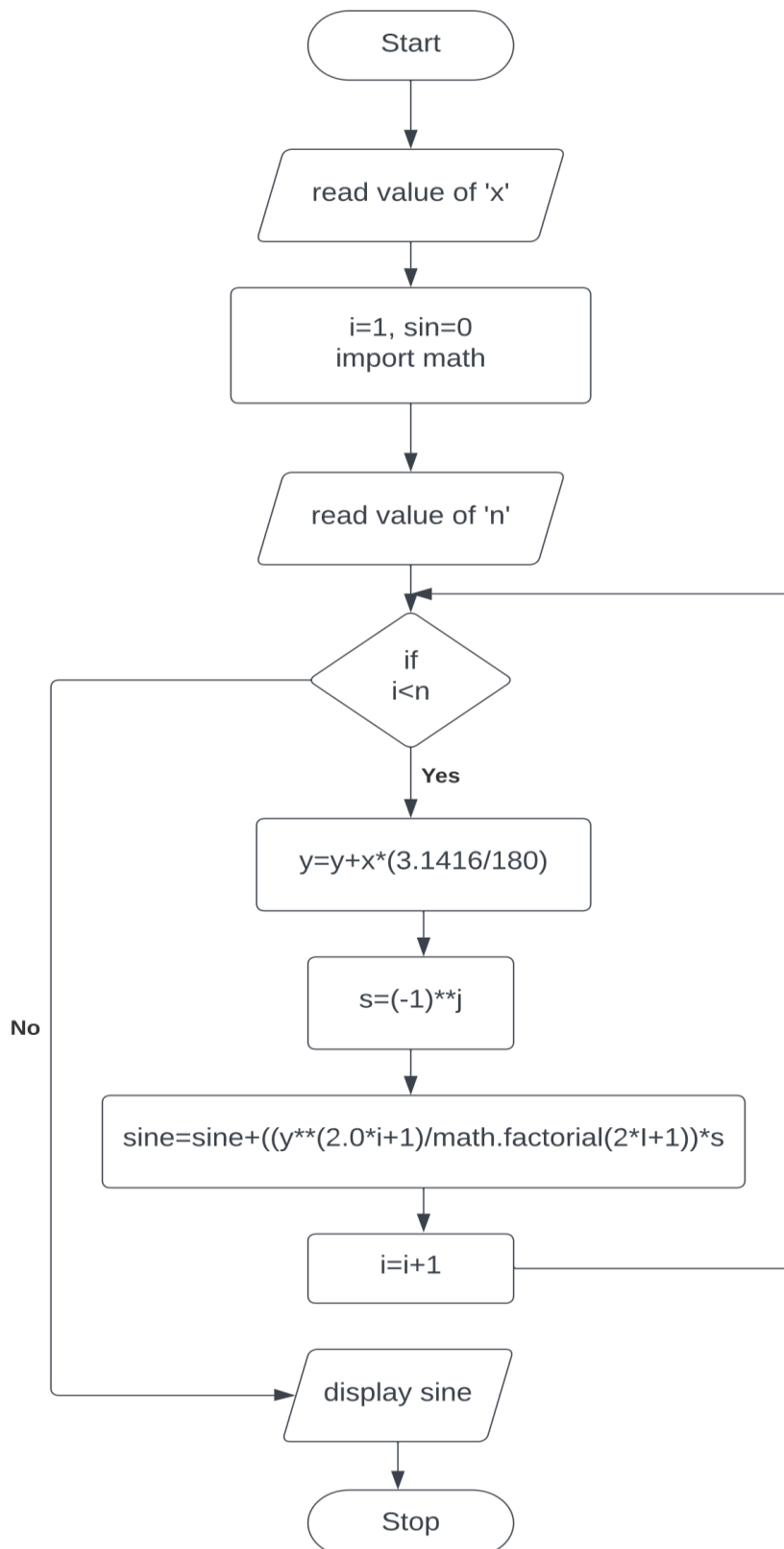   The algorithm and flowchart is written for the given problem.

Aim:

   To draw flowchart and write algorithm for the sine series.

Algorithm:

Step 1   : Start

Step 2   : read value of x

Step 3   : Compute i=1, sine=0, import math

Step 4   : Check if i<n

Step 4.1   : If Yes, then y=y+x*(3.1416/180), s=(-1)**j

Step 4.2   : Compute i=i+1, go to step 4

Step 5   : display sine and go to step 6

Step 6   : Stop

Flowchart:

22CSEA49 - NITHISH GOWTHAM.R

Pseudocode:

BEGIN

GET x

ASSIGN i=1, sine=0, import math

GET n

WHILE i<n

      COMPUTE y=y+x*(3.1416/180)

      COMPUTE s=(-1)**j

      COMPUTE sine=sine+((y**(2*i+1)/math.factorial(2*i+1))*s

      COMPUTE i=i+1

END WHILE

PRINT sine

END

Result:

    The algorithm and flowchart is written for the given problem.

22CSEA49 - NITHISH GOWTHAM.R