ALGORITHMS AND FLOWCHART (Day 1)

Q1. Write a c program to print the word 'ADAMAS UNIVERSITY'.

Ans:

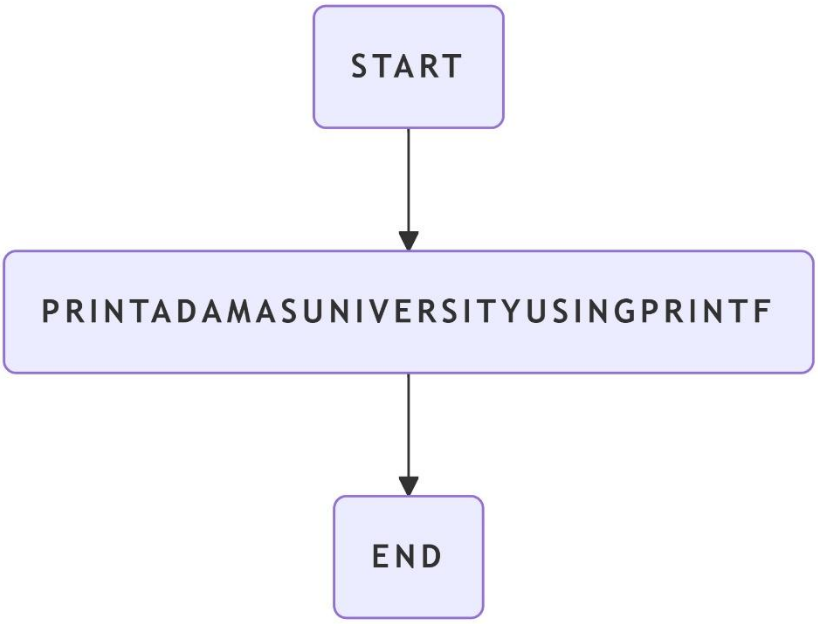
ALGORITHM

Step1: Start.

Step2: Print " ADAMAS UNIVERSITY" using 'printf'.

Step 3:End.

FLOWCHART



Q2. Write a program to add two float numbers.

Ans:

ALGORITHM

Step 1: start.

Step 2: Declare three variables a , b, c.

Step 3: Read the first number and store it in a.

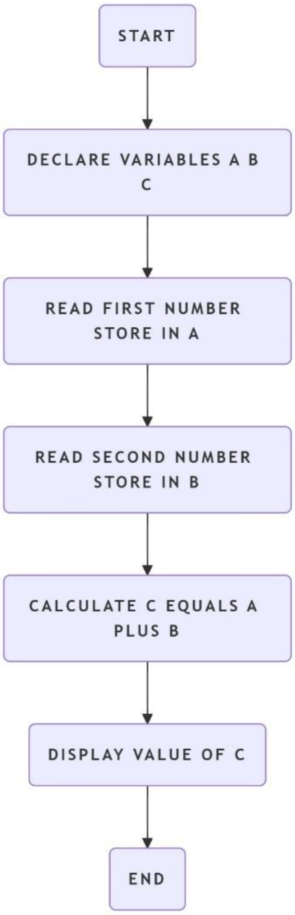
Step 4: Read the second number and store it in b.

Step 5: calculate   c=a+b;

Step 6: Display the value of c.

Step 7: End

FLOWCHART


START
DECLARE VARIABLES A B C
READ FIRST NUMBER STORE IN A
READ SECOND NUMBER STORE IN B
CALCULATE C EQUALS A PLUS B
DISPLAY VALUE OF C
END

Q3. Write a C program to substract two float integers.

Ans:

ALGORITHM

Step 1: Start.

Step 2: declare a the variables a , b, c.

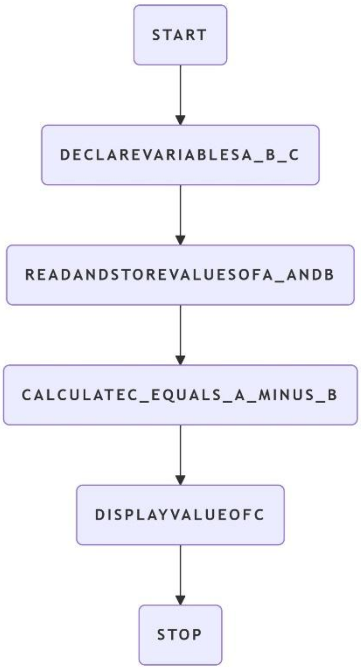
Step 3: read and store the value of two integers in a and b respectively.

Step 4: calculate c=a-b;

Step 5: Display the value of c.

Step 6: Stop.

FLOWCHART



Q4. Write a c program to write the multiplication table.

Ans:

ALGORITHM

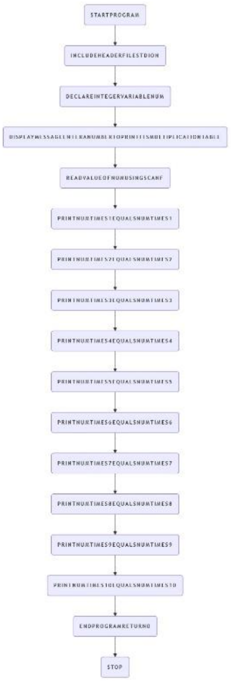
Step 1: Start the program.
Step 2: Include the header file <stdio.h>.
Step 3: Declare an integer variable num.

Step 4: Display a message — *“Enter a number to print its multiplication table:”*
Step 5: Read the value of num using scanf().
Step 6: Display the multiplication results as follows:

- Print num × 1 = num*1
- Print num × 2 = num*2
- Print num × 3 = num*3
- Print num × 4 = num*4
- Print num × 5 = num*5
- Print num × 6 = num*6
- Print num × 7 = num*7
- Print num × 8 = num*8
- Print num × 9 = num*9
- Print num × 10 = num*10

Step 7: End the program with return 0;.
Step 8: Stop.

FLOWCHART


STARTPROGRAM
INCLUDEHEADERFILESTDIOH
DECLAREINTEGERVARIABLENUM
DISPLAYMESSAGEENTERANUMBERTOPRINTITSMULTIPLICATIONTABLE
READVALUEOFNUMUSINGSCANF
PRINTNUMTIMES1EQUALSNUMTIMES1
PRINTNUMTIMES2EQUALSNUMTIMES2
PRINTNUMTIMES3EQUALSNUMTIMES3
PRINTNUMTIMES4EQUALSNUMTIMES4
PRINTNUMTIMES5EQUALSNUMTIMES5
PRINTNUMTIMES6EQUALSNUMTIMES6
PRINTNUMTIMES7EQUALSNUMTIMES7
PRINTNUMTIMES8EQUALSNUMTIMES8
PRINTNUMTIMES9EQUALSNUMTIMES9
PRINTNUMTIMES10EQUALSNUMTIMES10
ENDPROGRAMRETURN0
STOP

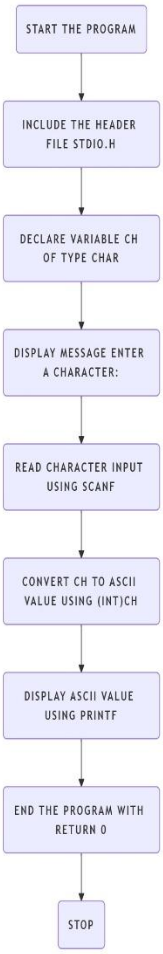
Q5.Write a c program to print the ASCII value of a character.

Ans:

Step 1: Start the program.
Step 2: Include the header file <stdio.h>.
Step 3: Declare a variable ch of type char.
Step 4: Display the message — *“Enter a character:”*

Step 5: Read the character input from the user using scanf("%c", &ch);
Step 6: Convert the character ch into its ASCII value using type casting (int)ch.
Step 7: Display the ASCII value using printf("The ASCII value of %c is %d", ch, (int)ch);
Step 8: End the program using return 0;.
Step 9: Stop.

FLOWCHART


START THE PROGRAM
INCLUDE THE HEADER FILE STDIO.H
DECLARE VARIABLE CH OF TYPE CHAR
DISPLAY MESSAGE ENTER A CHARACTER:
READ CHARACTER INPUT USING SCANF
CONVERT CH TO ASCII VALUE USING (INT)CH
DISPLAY ASCII VALUE USING PRINTF
END THE PROGRAM WITH RETURN 0
STOP

Q6. Write a c program to calculate the simple interest.

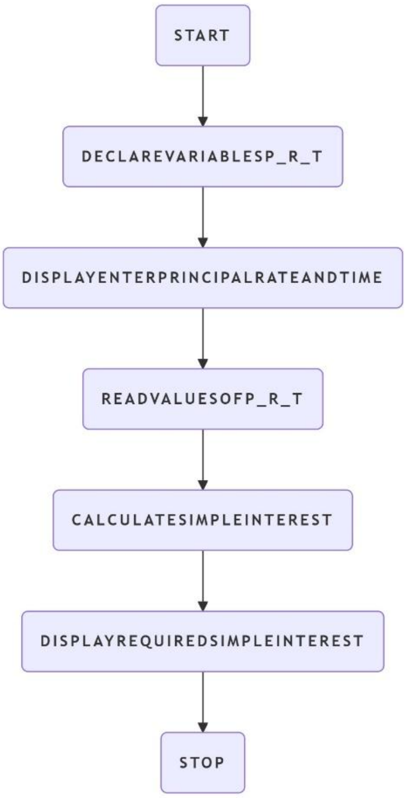
Ans:

ALGORITHM

1. **Start**

2. **Declare** three integer variables: p, r, and t
   (where p = principal amount, r = rate of interest, and t = time)
3. **Display** the message:
   "Enter the principal, rate of interest and time"
4. **Read** the values of p, r, and t from the user.
5. **Calculate** the simple interest using the formula:
   si = (p * r * t) / 100
6. **Display** the result:
   "Required simple interest: si"
7. **Stop**

FLOWCHART

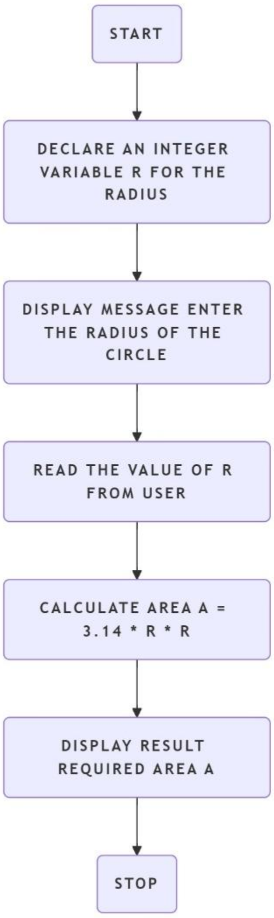


Q7. Write a c program to calculate the area of a circle.

Ans:

ALGORITHM

1. **Start**
2. **Declare** an integer variable r (for the radius of the circle).
3. **Display** the message:
   "Enter the radius of the circle"
4. **Read** the value of r from the user.
5. **Calculate** the area using the formula:
   a = 3.14 * r * r
6. **Display** the result:
   "Required area: a"
7. **Stop**


START
DECLARE AN INTEGER VARIABLE R FOR THE RADIUS
DISPLAY MESSAGE ENTER THE RADIUS OF THE CIRCLE
READ THE VALUE OF R FROM USER
CALCULATE AREA A = 3.14 * R * R
DISPLAY RESULT REQUIRED AREA A
STOP

Q8.Write a c program to convert frahreinheit to celcius and vice versa.

Ans:

ALGORITHM

1.**Start**

2. **Declare** four float variables: f1, f2, c1, and c2.

3. **Display** the message:
"Enter temperature in Fahrenheit:"

4. **Read** the value of f1 from the user.

5. **Display** the message:
"Enter temperature in Celsius:"

6.**Read** the value of c1 from the user.

7. **Calculate** the Celsius equivalent of Fahrenheit using the formula:
c2 = (f1 - 32) * 5 / 9

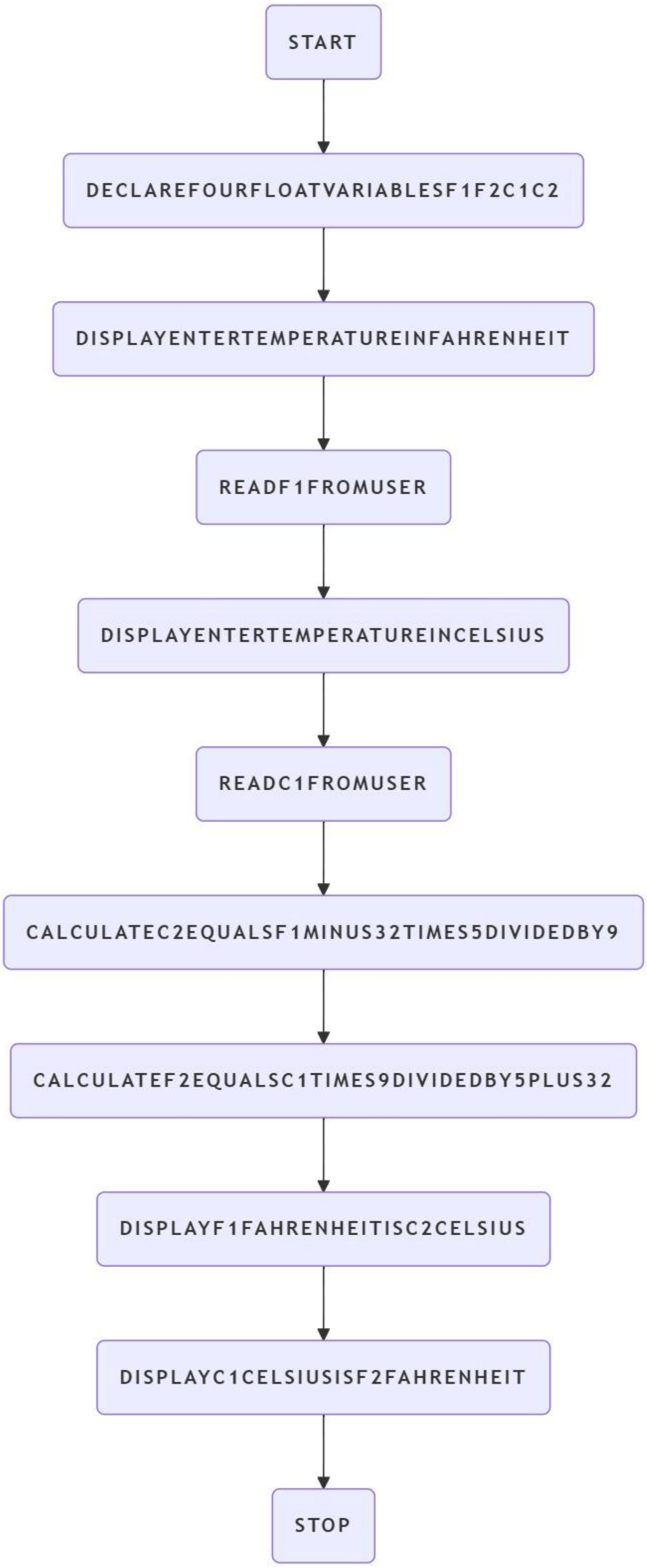
8.**Calculate** the Fahrenheit equivalent of Celsius using the formula:
f2 = (c1 * 9 / 5) + 32

9.**Display** the result:
f1 Fahrenheit is c2 Celsius

10. **Display** the result:
c1 Celsius is f2 Fahrenheit

11. **Stop**

FLOWCHART


START
DECLAREFOURFLOATVARIABLESF1F2C1C2
DISPLAYENTERTEMPERATUREINFAHRENHEIT
READF1FROMUSER
DISPLAYENTERTEMPERATUREINCELSIUS
READC1FROMUSER
CALCULATEC2EQUALSF1MINUS32TIMES5DIVIDEDBY9
CALCULATEF2EQUALSC1TIMES9DIVIDEDBY5PLUS32
DISPLAYF1FAHRENHEITISC2CELSIUS
DISPLAYC1CELSIUSISF2FAHRENHEIT
STOP

Q9. Write a c program to swap two numbers using   a third variable.

Ans:

ALGORITHM

1. **Start**

2.**Declare** three integer variables: a, b, and c.

3.**Display** the message:
"Enter two numbers"

4.**Read** the values of a and b from the user.

5.**Store** the value of b in c.
→ c = b

6.**Assign** the value of a to b.
→ b = a

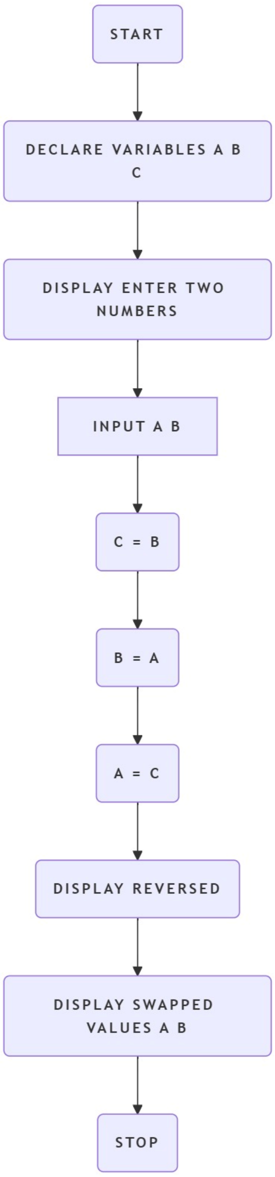
7.**Assign** the value of c to a.
→ a = c

8.**Display** the message:
"Reversed"

9. **Display** the swapped values of a and b.

10.**Stop**

FLOWCHART



Q10. Write a c program to swap two numbers without using a third variable.

Ans:

ALGORITHM

1. **Start**

2.**Declare** three integer variables: a, b, and c (though c is unused here, you can skip declaring it).

3.**Display** the message:
"Enter two numbers"

4.**Read** the values of a and b from the user.

5.**Swap** the numbers using arithmetic operations:

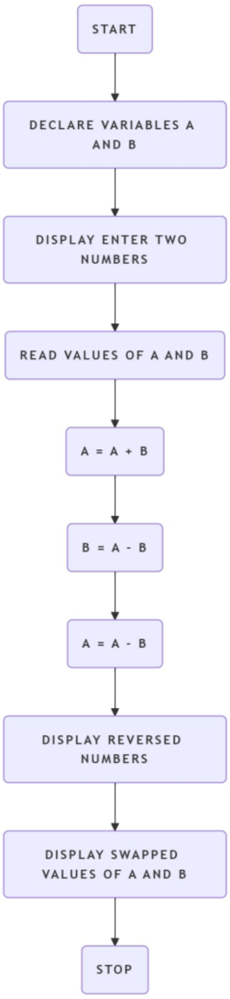
- a = a + b
- b = a - b
- a = a - b

6.**Display** the message:
"Reversed numbers"

7.**Display** the swapped values of a and b.

8. **Stop**

FLOWCHART


START
DECLARE VARIABLES A AND B
DISPLAY ENTER TWO NUMBERS
READ VALUES OF A AND B
A = A + B
B = A - B
A = A - B
DISPLAY REVERSED NUMBERS
DISPLAY SWAPPED VALUES OF A AND B
STOP

Q11.Write a c program to find the last digit of an integer (i) using modulus operator (ii) without using modulus operator.

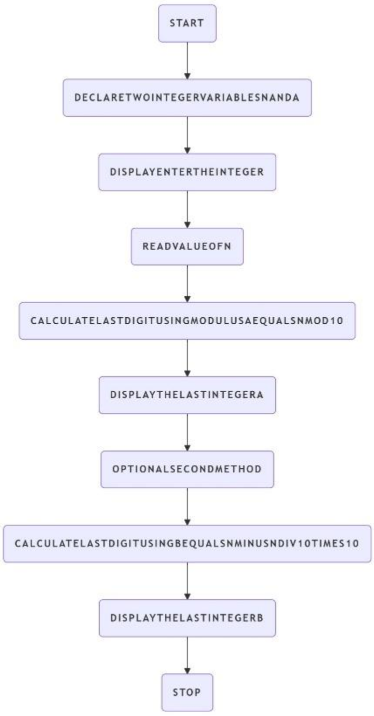
Ans:

ALGORITHM

1. **Start**
2. **Declare** two integer variables: n and a.
3. **Display** the message:
   "Enter the integer"
4. **Read** the value of n from the user.
5. **Find the last digit** of the number using the modulus operator:
   a = n % 10
6. **Display** the message and result:
   "The last integer: a"
7. **(Optional second method)**
   Compute the last digit again using:
   b = n - (n / 10) * 10
8. **Display** the message and result:
   "The last integer: b"
9. **Stop**

10. **Start**
11. **Declare** two integer variables: `n` and `a`.
12. **Display** the message:
    `"Enter the integer"`
13. **Read** the value of `n` from the user.
14. **Find the last digit** of the number using the modulus operator:
    `a = n % 10`
15. **Display** the message and result:
    `"The last integer: a"`
16. **(Optional second method)**
    Compute the last digit again using:
    `b = n - (n / 10) * 10`
17. **Display** the message and result:
    `"The last integer: b"`
18. **Stop**

FLOWCHART


START
DECLARETWOINTEGERVARIABLESNANDA
DISPLAYENTERTHEINTEGER
READVALUEOFN
CALCULATELASTDIGITUSINGMODULUSAEQUALSNMOD10
DISPLAYTHELASTINTEGERA
OPTIONALSECONDMETHOD
CALCULATELASTDIGITUSINGBEQUALSNMINUSNDIV10TIMES10
DISPLAYTHELASTINTEGERB
STOP

Q12. Write a c program to find out the compound interest.

Ans:

ALGORITHM

1. **Start**

2.**Declare** the following variables of type double:
principal, rate, time, n, amount, and compoundInterest.

3.**Display** the message:
"Enter the principal amount:"

4.**Read** the value of principal.

5.**Display** the message:
"Enter the annual interest rate (in percentage):"

6.**Read** the value of rate.

7.**Display** the message:
"Enter the time (in years):"

8.**Read** the value of time.

9.**Display** the message:
"Enter the number of times interest is compounded per year:"

10.**Read** the value of n.

11.**Convert** the annual interest rate into decimal form:
rate = rate / 100

12. **Calculate** the total amount using the compound interest formula:
amount = principal * (1 + rate / n)^(n * time)
(Use the pow() function for exponentiation.)

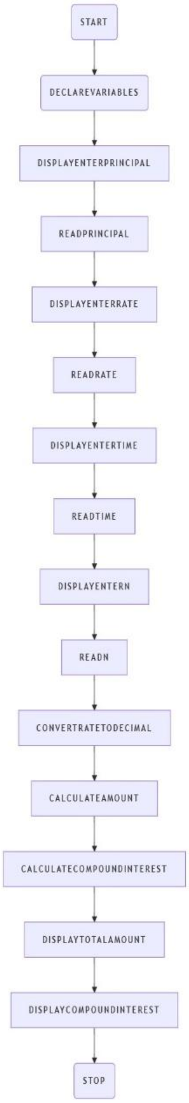
13. **Calculate** the compound interest:
compound Interest = amount - principal

14.**Display** the total amount after the given time:
"Total Amount after time years: amount"

15.**Display** the compound interest earned:
"Compound Interest earned: compound Interest"

16. **Stop**

FLOWCHART



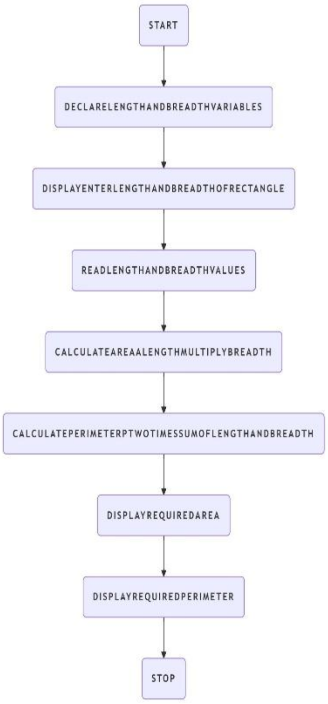
Q13.Write a c program to find out the area and perimeter of a rectangle.

Ans:

ALGORITHM

1. **Start**
2. **Declare** two integer variables: l (length) and b (breadth).
3. **Display** the message:
   "Enter the length and breadth of a rectangle"
4. **Read** the values of l and b from the user.
5. **Calculate** the area of the rectangle using the formula:
   a = l * b
6. **Calculate** the perimeter of the rectangle using the formula:
   p = 2 * (l + b)
7. **Display** the result for area:
   "Required area: a"
8. **Display** the result for perimeter:
   "Required perimeter: p"
9. **Stop**

FLOWCHART


START
DECLARELENGTHANDBREADTHVARIABLES
DISPLAYENTERLENGTHANDBREADTHOFRECTANGLE
READLENGTHANDBREADTHVALUES
CALCULATEAREAALENGTHMULTIPLYBREADTH
CALCULATEPERIMETERPTWOTIMESSUMOFLENGTHANDBREADTH
DISPLAYREQUIREDAREA
DISPLAYREQUIREDPERIMETER
STOP

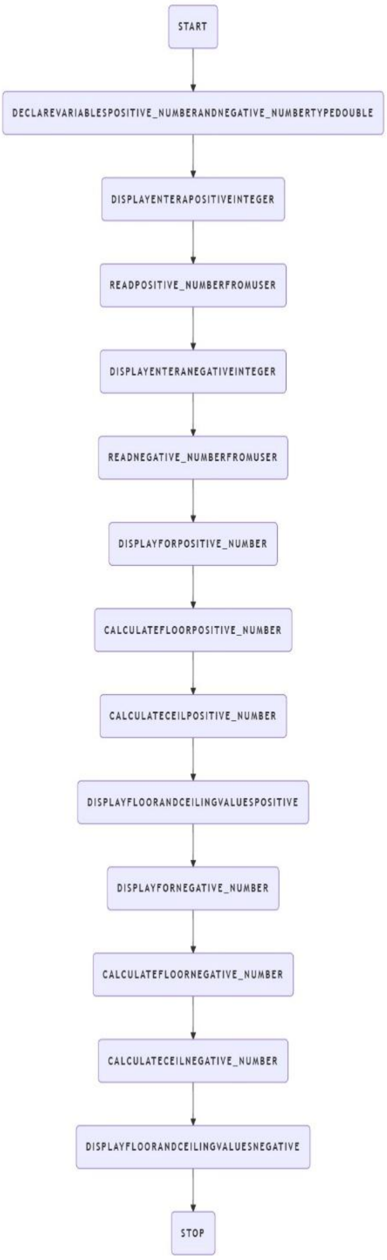
Q14.Write a c program to find out the floor and ceil value of positive and negative integers.

Ans:

ALGORITHM

1. **Start**
2. **Declare** two variables of type double:
   positive_number and negative_number.
3. **Display** the message:
   "Enter a positive integer:"
4. **Read** the value of positive_number from the user.
5. **Display** the message:
   "Enter a negative integer:"
6. **Read** the value of negative_number from the user.
7. **For the positive number:**
   - **Display** "For the positive number:"
   - **Calculate** the floor value using:
     floor(positive_number)
   - **Calculate** the ceiling value using:
     ceil(positive_number)
   - **Display** both the floor and ceiling values.
8. **For the negative number:**
   - **Display** "For the negative number:"
   - **Calculate** the floor value using:
     floor(negative_number)
   - **Calculate** the ceiling value using:
     ceil(negative_number)
   - **Display** both the floor and ceiling values.
9. **Stop**

FLOWCHART


START
DECLAREVARIABLESPOSITIVE_NUMBERANDNEGATIVE_NUMBERTYPEDOUBLE
DISPLAYENTERAPOSITIVEINTEGER
READPOSITIVE_NUMBERFROMUSER
DISPLAYENTERANEGATIVEINTEGER
READNEGATIVE_NUMBERFROMUSER
DISPLAYFORPOSITIVE_NUMBER
CALCULATEFLOORPOSITIVE_NUMBER
CALCULATECEILPOSITIVE_NUMBER
DISPLAYFLOORANDCEILINGVALUESPOSITIVE
DISPLAYFORNEGATIVE_NUMBER
CALCULATEFLOORNEGATIVE_NUMBER
CALCULATECEILNEGATIVE_NUMBER
DISPLAYFLOORANDCEILINGVALUESNEGATIVE
STOP

Q15.Write a c program to find out the roots of a quadratic equation but only the positive part.

Ans:

ALGORITHM

1. **Start**
2. **Declare** variables of type double:
   a, b, c, discriminant, root1, and root2.

3. **Display** the message:
   "Enter coefficients a, b, and c for the equation $ax^2 + bx + c = 0$"

4. **Read** the values of a, b, and c from the user.

5. **Calculate** the discriminant using the formula:
   discriminant = (b * b) - (4 * a * c)

6. **Check** the condition:

   - **If** discriminant >= 0, then proceed to find real roots.

   - **Else**, display:
     "The equation has no real roots, and therefore no positive roots."
     and go to step 12.

7. **Calculate** the two real roots using the quadratic formula:

   - root1 = (-b + sqrt(discriminant)) / (2 * a)

   - root2 = (-b - sqrt(discriminant)) / (2 * a)

8. **Display** the message:
   "Positive roots are:"

9. **Check for positive roots:**

   - **If** root1 > 0, display root1.

   - **If** root2 > 0, display root2.

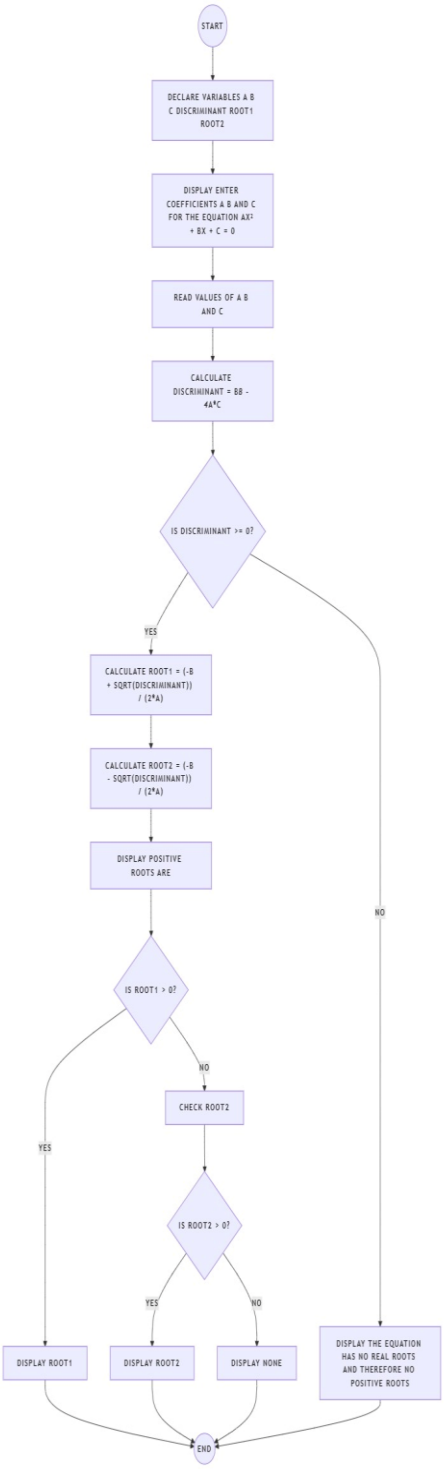
   - **If both** root1 <= 0 **and** root2 <= 0, display "None".

10. **Display** the positive roots (if any).

11. **If discriminant < 0**, display:
    "No real or positive roots."

12. **Stop**

FLOWCHART


START
DECLARE VARIABLES A B C DISCRIMINANT ROOT1 ROOT2
DISPLAY ENTER COEFFICIENTS A B AND C FOR THE EQUATION AX² + BX + C = 0
READ VALUES OF A B AND C
CALCULATE DISCRIMINANT = BB - 4A*C
IS DISCRIMINANT >= 0?
YES
CALCULATE ROOT1 = (-B + SQRT(DISCRIMINANT)) / (2*A)
CALCULATE ROOT2 = (-B - SQRT(DISCRIMINANT)) / (2*A)
DISPLAY POSITIVE ROOTS ARE
NO
IS ROOT1 > 0?
NO
CHECK ROOT2
YES
IS ROOT2 > 0?
YES
NO
DISPLAY ROOT1
DISPLAY ROOT2
DISPLAY NONE
DISPLAY THE EQUATION HAS NO REAL ROOTS AND THEREFORE NO POSITIVE ROOTS
END

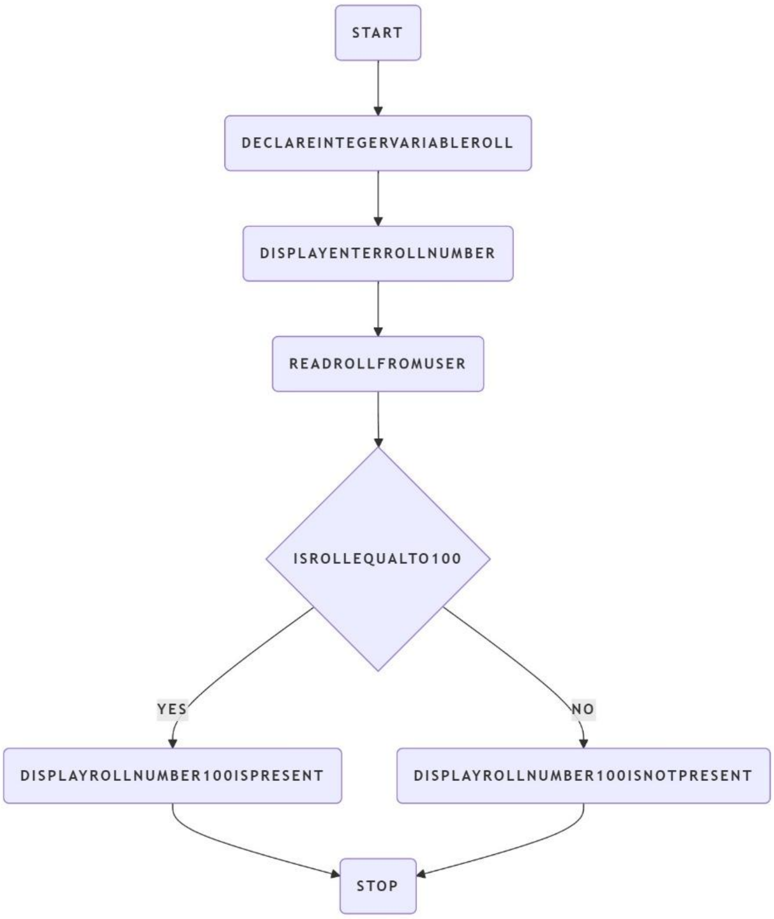
Q16.Write a c program to check whether roll no. 100 is present or not using if only.

Ans:

ALGORITHM

1. **Start**
2. **Declare** an integer variable roll.
3. **Display** the message:
   "Enter roll number:"
4. **Read** the value of roll from the user.
5. **Check** the condition:
   - **If** roll == 100, then
     → Display "Roll number 100 is present."
6. **Check** the next condition:
   - **If** roll != 100, then
     → Display "Roll number 100 is not present."
7. **Stop**

FLOWCHART


START
DECLAREINTEGERVARIABLEROLL
DISPLAYENTERROLLNUMBER
READROLLFROMUSER
ISROLLEQUALTO100
YES
NO
DISPLAYROLLNUMBER100ISPRESENT
DISPLAYROLLNUMBER100ISNOTPRESENT
STOP

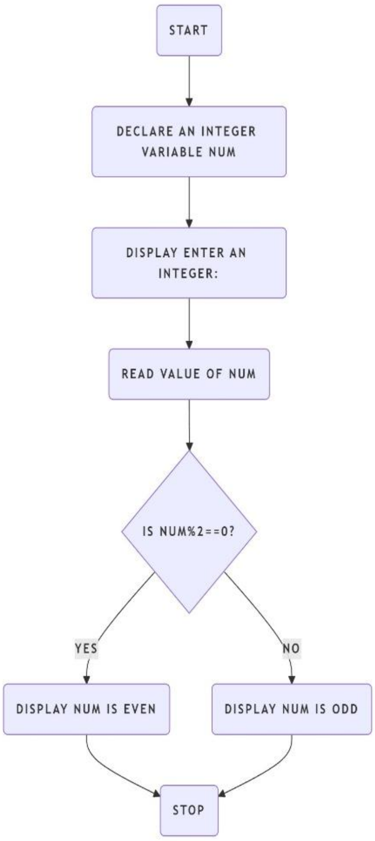
Q17.Write a c program to check an integer whether its odd or even.

Ans:

ALGORITHM

1. **Start**
2. **Declare** an integer variable num.
3. **Display** the message:
   "Enter an integer:"
4. **Read** the value of num from the user.
5. **Check** the condition:
   - **If** num % 2 == 0, then
     → Display "<num> is Even."
6. **Else**,
   → Display "<num> is Odd."
7. **Stop**

FLOWCHART



Q18.Write a c program to check the greatest among the two numbers.

Ans:

ALGORITHM

1. **Start**

2.**Declare** two integer variables: a and b.

3.**Display** the message:
"Enter two numbers:"

4.**Read** the values of a and b from the user.

5. **Compare** the two numbers:

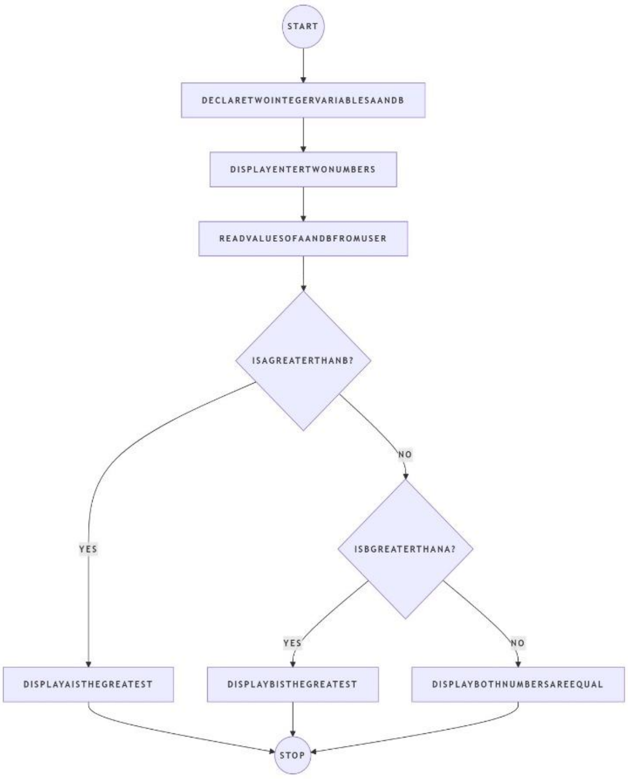
- **If** a > b, then
  → Display "<a> is the greatest."

- **Else if** b > a, then
  → Display "<b> is the greatest."

- **Else** (i.e., a == b),
  → Display "Both numbers are equal."

6.**Stop**

FLOWCHART



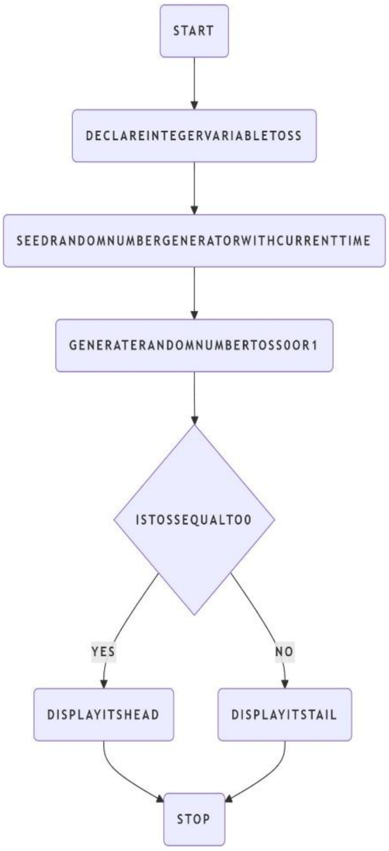
Q19.A coin has been tossed once . Write a c program to check whether its head or tails.

Ans:

ALGORITHM

1. **Start**
2. **Declare** an integer variable toss.
3. **Seed** the random number generator using the current time:
   srand(time(0))
4. **Generate** a random number toss as either 0 or 1:
   toss = rand() % 2
5. **Check** the value of toss:
   - **If** toss == 0, then
     → Display "It's Head!"
   - **Else** (i.e., toss == 1),
     → Display "It's Tail!"
6. **Stop**

FLOWCHART


START
DECLAREINTEGERVARIABLETOSS
SEEDRANDOMNUMBERGENERATORWITHCURRENTTIME
GENERATERANDOMNUMBERTOSS0OR1
ISTOSSEQUALTO0
YES
NO
DISPLAYITSHEAD
DISPLAYITSTAIL
STOP

Q20.Write a c program to check whether a number is a whole number or not.

Ans:

ALGORITHM

1. **Start**

2. **Declare** an integer variable num.

3. **Display** the message:
"Enter a number:"

4. **Read** the value of num from the user.

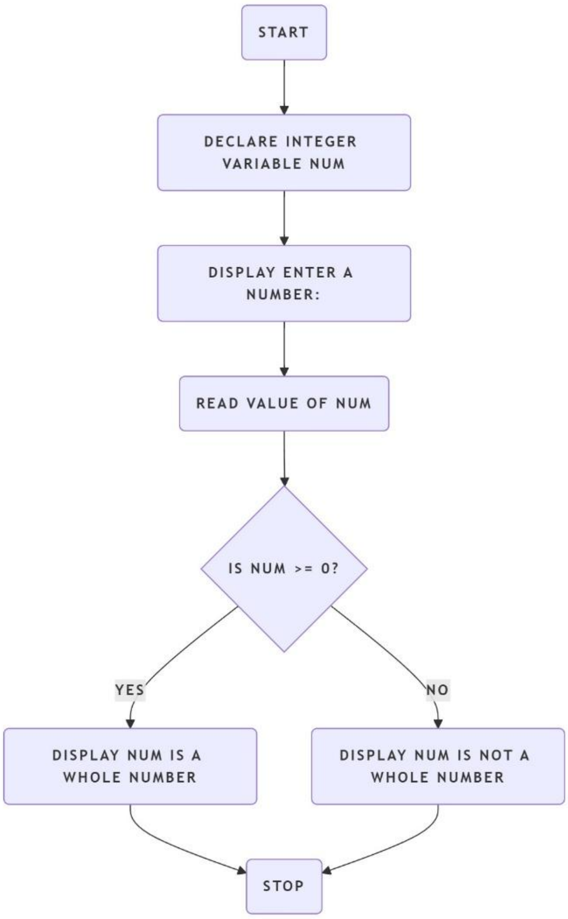
5. **Check** the condition:

- **If** num >= 0, then
→ Display "<num> is a whole number"

- **Else** (i.e., num < 0),
→ Display "<num> is not a whole number"

6.**Stop**

FLOWCHART


START
DECLARE INTEGER VARIABLE NUM
DISPLAY ENTER A NUMBER:
READ VALUE OF NUM
IS NUM >= 0?
YES
NO
DISPLAY NUM IS A WHOLE NUMBER
DISPLAY NUM IS NOT A WHOLE NUMBER
STOP

Q21. Write a c program to check whether it is a leap year or not using ladder if else.

Ans:

ALGORITHM

1. **Start**

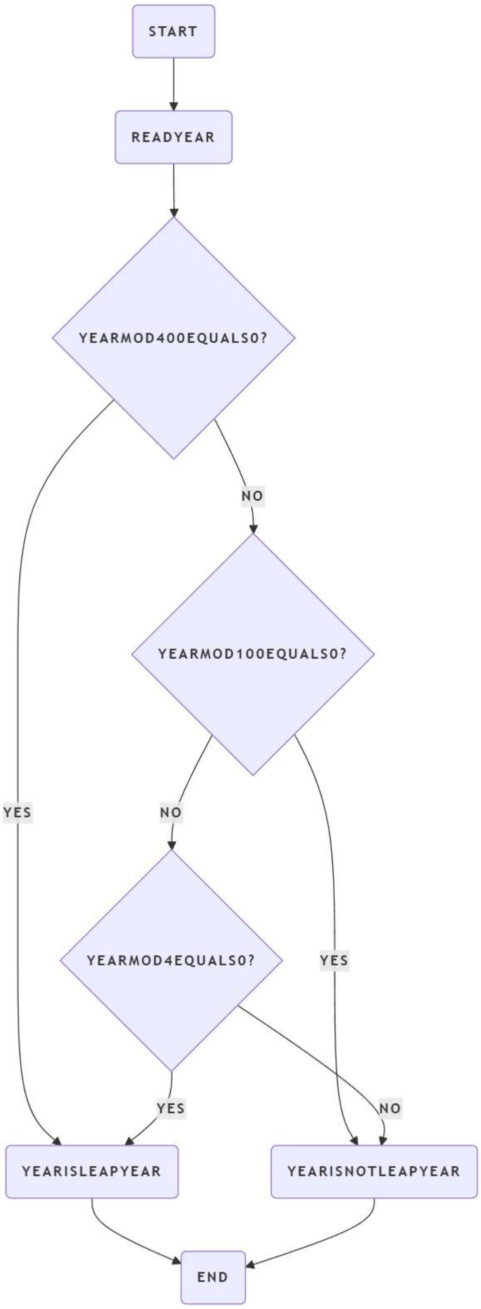
2.**Input**: Read an integer year from the user.

3.**Process**:

- If year % 400 == 0, then
  - Print "year is a Leap Year."
- Else if year % 100 == 0, then
  - Print "year is Not a Leap Year."
- Else if year % 4 == 0, then
  - Print "year is a Leap Year."
- Else
  - Print "year is Not a Leap Year."

4. **End**

FLOWCHART



Q22. Write a c program to find the greatest among three numbers.

Ans:

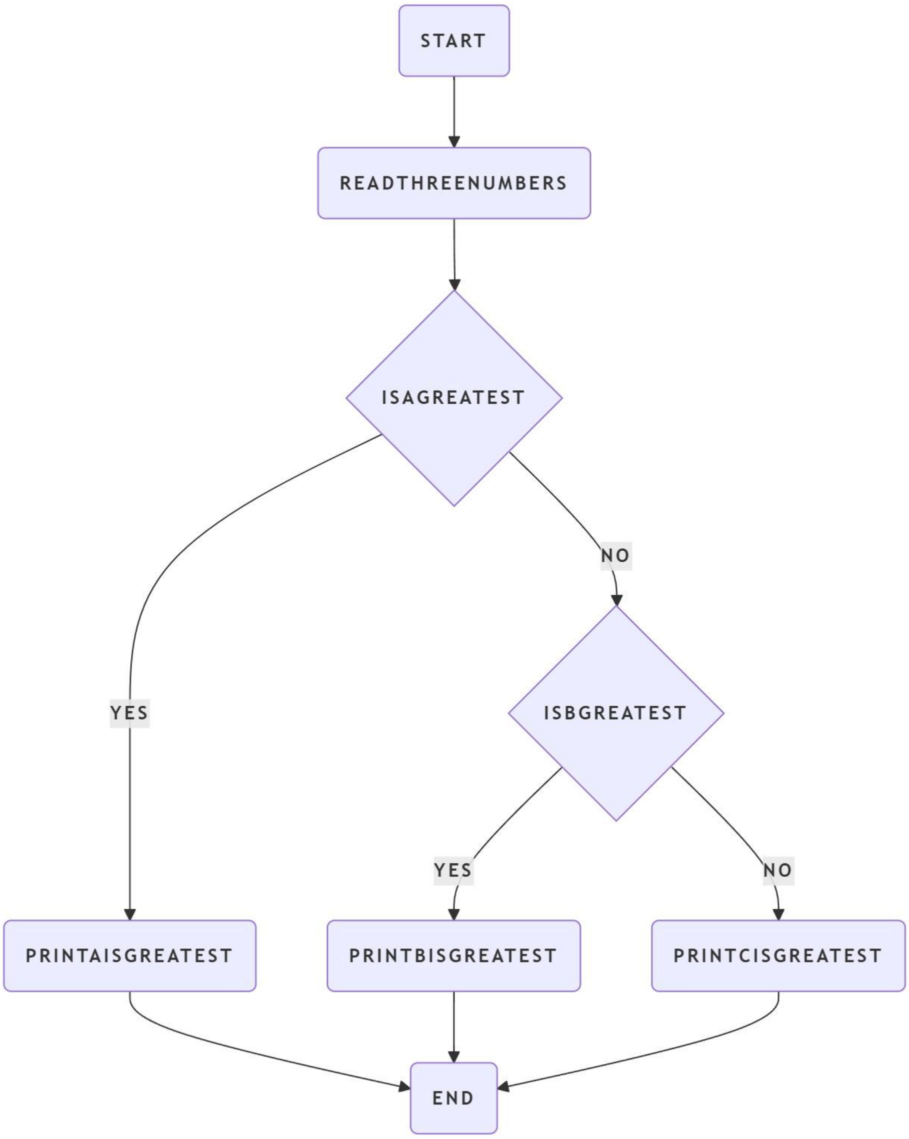
ALGORITHM

1. **Start**
2. **Input**: Read three numbers a, b, and c from the user.
3. **Process**:

- If a >= b **and** a >= c, then
  - Print "a is the greatest."
- Else if b >= a **and** b >= c, then
  - Print "b is the greatest."
- Else
  - Print "c is the greatest."

4. **End**

FLOWCHART



Q23.Write a c program r check whether the letter is vowel or consonant using ladder if else.

Ans:

ALGORITHM

1. **Start**

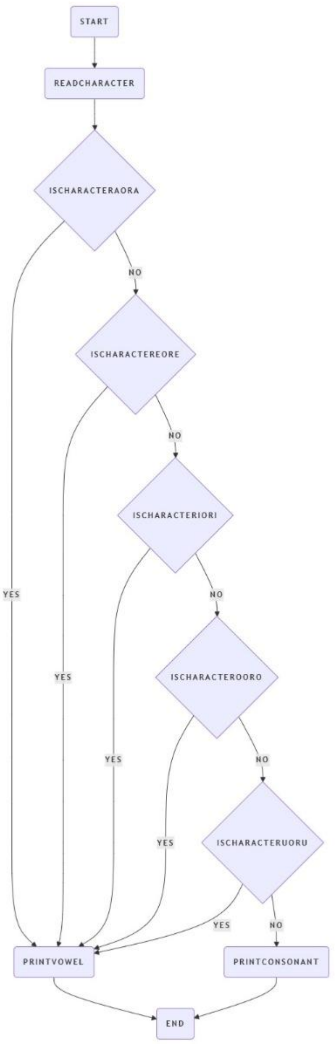
2. **Input**: Read a character ch from the user.

3. **Process**:

- If ch is 'a' or 'A', then
    - Print "ch is a vowel."
- Else if ch is 'e' or 'E', then
    - Print "ch is a vowel."
- Else if ch is 'i' or 'I', then
    - Print "ch is a vowel."
- Else if ch is 'o' or 'O', then
    - Print "ch is a vowel."
- Else if ch is 'u' or 'U', then
    - Print "ch is a vowel."
- Else
    - Print "ch is a consonant."

5. **End**

FLOWCHART



Q24. Write a c program to find the size of an integer , float, double,char.

Ans:

ALGORITHM

1.**Start**

2.**Process**:

- Use sizeof(int) to find the size of an integer.
- Use sizeof(float) to find the size of a float.
- Use sizeof(double) to find the size of a double.
- Use sizeof(char) to find the size of a character.

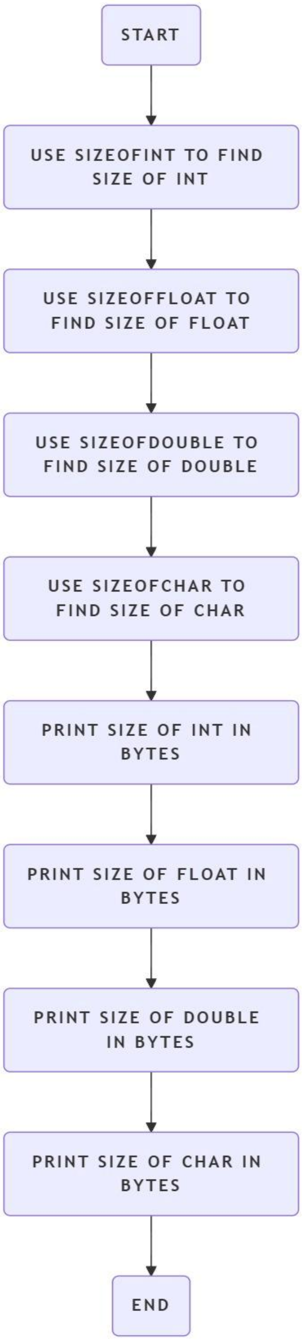
3. **Output**:

- Print the size of int in bytes.

- Print the size of float in bytes.
- Print the size of double in bytes.
- Print the size of char in bytes.

4. **End**

FLOWCHART


START
USE SIZEOFINT TO FIND SIZE OF INT
USE SIZEOFFLOAT TO FIND SIZE OF FLOAT
USE SIZEOFDOUBLE TO FIND SIZE OF DOUBLE
USE SIZEOFCHAR TO FIND SIZE OF CHAR
PRINT SIZE OF INT IN BYTES
PRINT SIZE OF FLOAT IN BYTES
PRINT SIZE OF DOUBLE IN BYTES
PRINT SIZE OF CHAR IN BYTES
END

Q25.Write a c program to check maximum among three numbers using nested if ele.

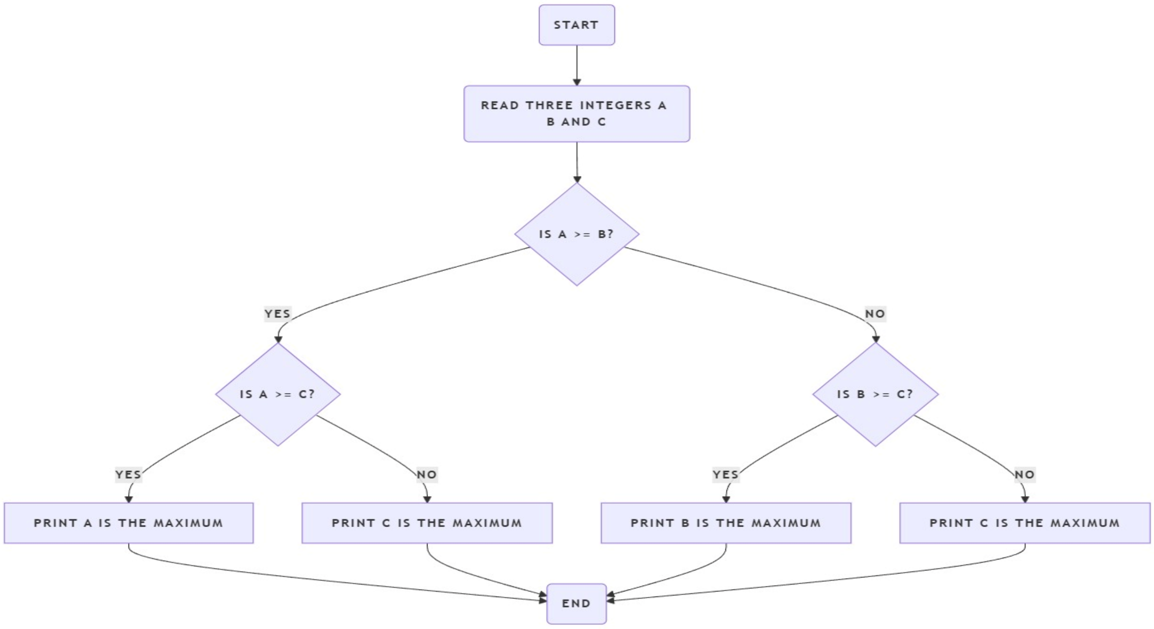
Ans:

ALGORITHM

Algorithm: Find the Maximum of Three Numbers (Nested If)

1. **Start**
2. **Input**: Read three integers a, b, and c from the user.
3. **Process**:
    - If a >= b then
        - If a >= c then
            - Print "a is the maximum."
        - Else
            - Print "c is the maximum."
    - Else (a < b)
        - If b >= c then
            - Print "b is the maximum."
        - Else
            - Print "c is the maximum."
4. **End**

FLOWCHART



Q26.Write a c program to create a simple calculator using switch case.

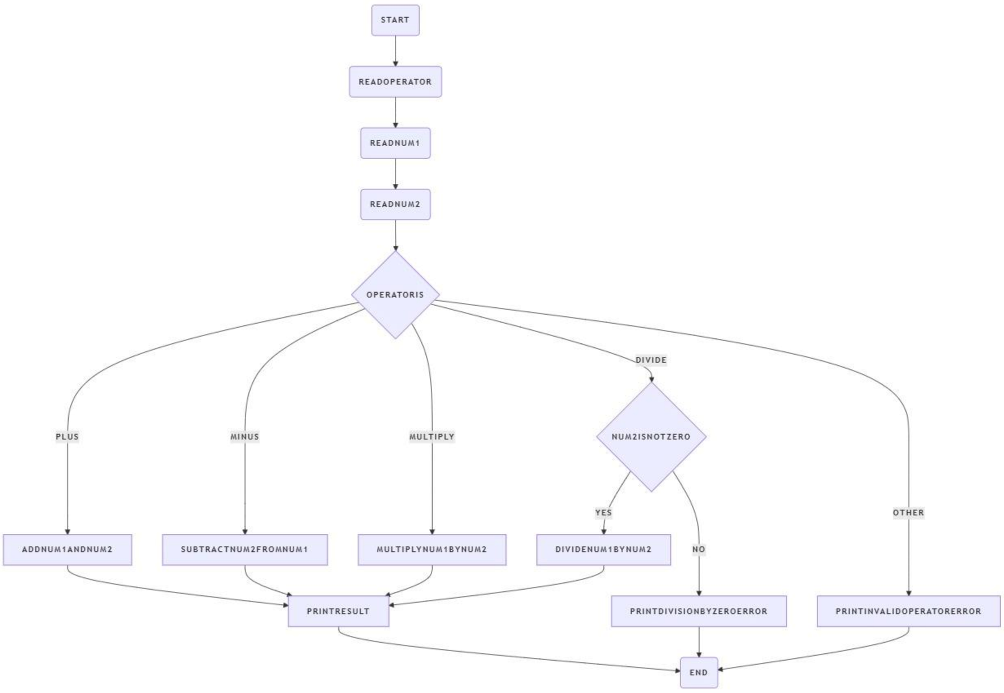
Ans:

ALGORITHM

1. **Start**
2. **Input**:
   - Read an operator op (+, -, *, /) from the user.
   - Read two numbers num1 and num2 from the user.

3. **Process** (Decision based on operator):
    - **Switch op:**
        - Case '+':
            - result = num1 + num2
            - Print result
        - Case '-':
            - result = num1 - num2
            - Print result
        - Case '*':
            - result = num1 * num2
            - Print result
        - Case '/':
            - If num2 != 0 then
                - result = num1 / num2
                - Print result
            - Else
                - Print "Error! Division by zero."
                - Stop program
        - Default:
            - Print "Error! Invalid operator."
4. **End**

FLOWCHART



Q27. Write a c program using switch case to check whether a given alphabet is a vowel or constant.

Ans:

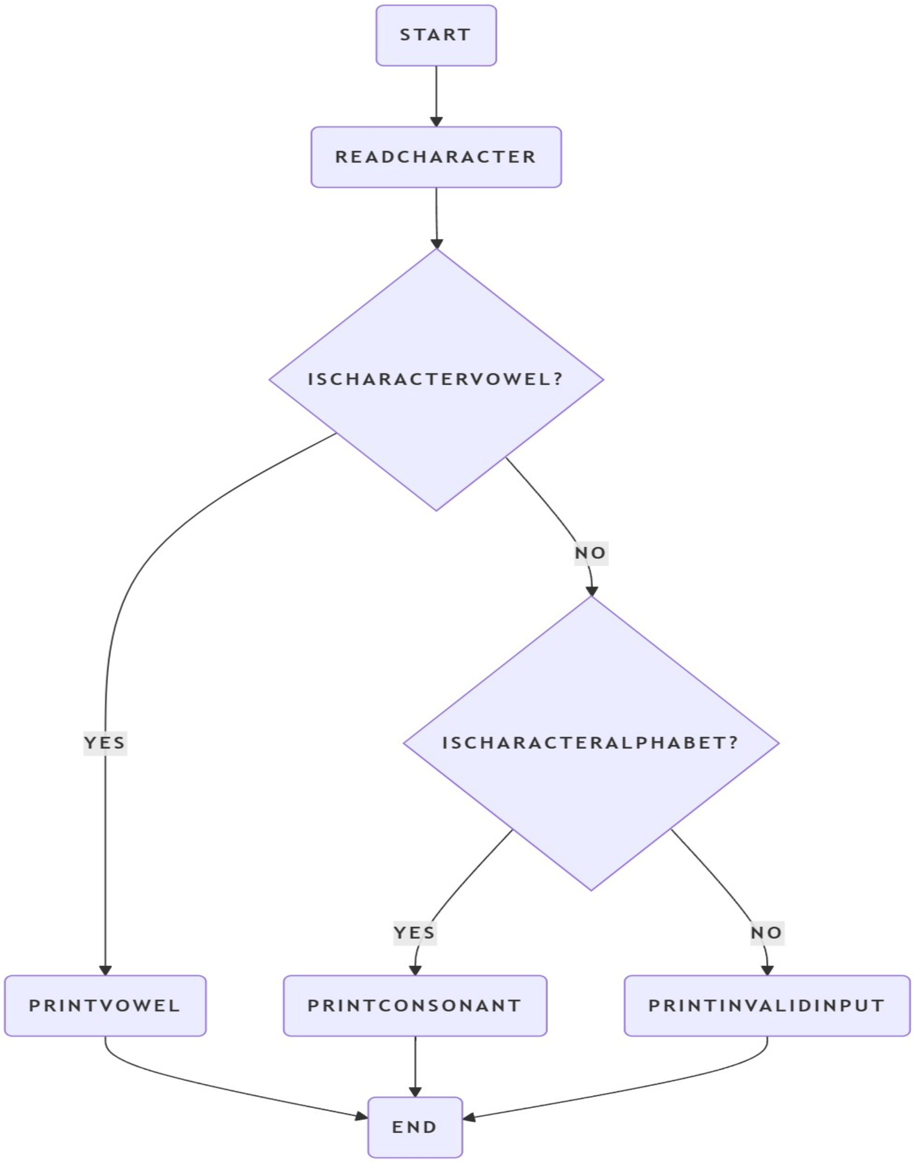
ALGORITHM

1. **Start**
2. **Input**: Read a character ch from the user.
3. **Process** (Using switch on ch):

- **Case 'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U'**:
    - Print "ch is a vowel."
- **Default**:
    - If ch is between 'a' and 'z' **or** between 'A' and 'Z' then
        - Print "ch is a consonant."
    - Else
        - Print "Invalid input! Please enter an alphabet."

4. **End**

FLOWCHART



Q28. Write a c program to find out the salary of an employee using switch case.

Ans:

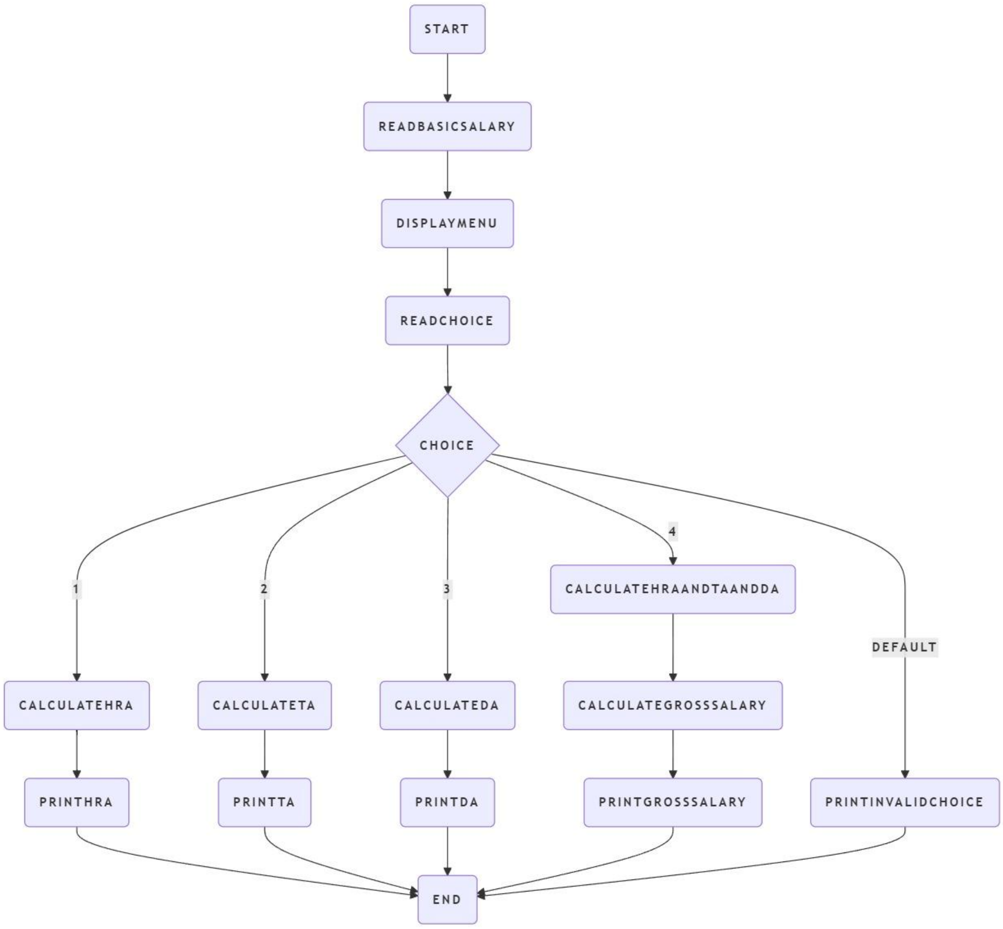
ALGORITHM

1. **Start**
2. **Input**:
   - Read basic salary from the user.
3. **Display Menu**:
   -
     1. Calculate HRA (20% of Basic)
   -
     2. Calculate TA (10% of Basic)
   -
     3. Calculate DA (5% of Basic)
   -
     4. Calculate Gross Salary
4. **Input**: Read choice from the user.
5. **Process** (Using switch on choice):
   - **Case 1**:
     - hra = 0.20 * basic
     - Print HRA
   - **Case 2**:
     - ta = 0.10 * basic
     - Print TA
   - **Case 3**:
     - da = 0.05 * basic
     - Print DA
   - **Case 4**:
     - hra = 0.20 * basic

        - ta = 0.10 * basic
        - da = 0.05 * basic
        - gross = basic + hra + ta + da
        - Print Gross Salary
    - **Default**:
        - Print "Invalid choice!"

6. **End**

FLOWCHART



Q29.write a C program using switch case to calculate the electricity bill in a house.

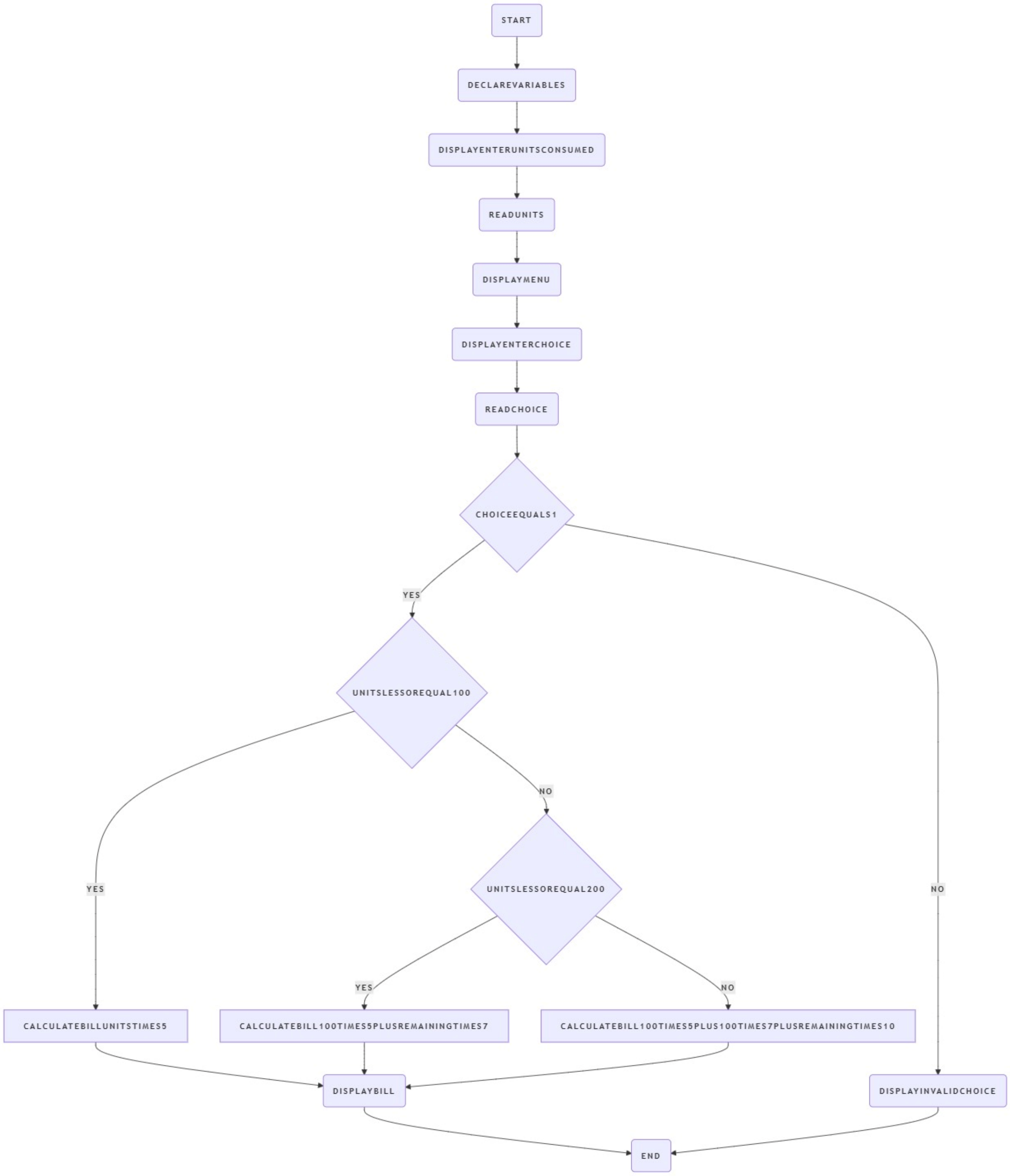
Ans:

ALGORITHM

Algorithm: Electricity Bill Calculation

1. **Start**
2. **Declare** variables:
    - units (integer)
    - choice (integer)
    - bill (double, initialize to 0)
3. **Display** message: “Enter units consumed:”
4. **Read** the value of units
5. **Display** the menu:
6. Menu:
7. 1. Calculate bill
8. **Display** message: “Enter your choice:”
9. **Read** the value of choice
10. **Use switch(choice)** to decide:
    - **Case 1:**
        - If units <= 100
→ bill = units * 5
        - Else if units <= 200
→ bill = (100 * 5) + (units - 100) * 7
        - Else
→ bill = (100 * 5) + (100 * 7) + (units - 200) * 10
        - **Display:** “Electricity Bill = ₹bill”
        - **Break**
    - **Default:**
        - **Display:** “Invalid choice!”
11. **End**

FLOWCHART



Q30.Write a c program using switch case to display the grade of a student based on their marks.

a. 95 and above → Grade A+
b. 90 and 94 →Grade A

c. 80 and 89→ Grade B
d. 70 and 79→ Grade C
e. 70 and below→Grade F

Ans:

ALGORITHM

1. **Start**

2.**Declare** an integer variable marks

3.**Display** message: “Enter marks:”

4.**Read** the value of marks

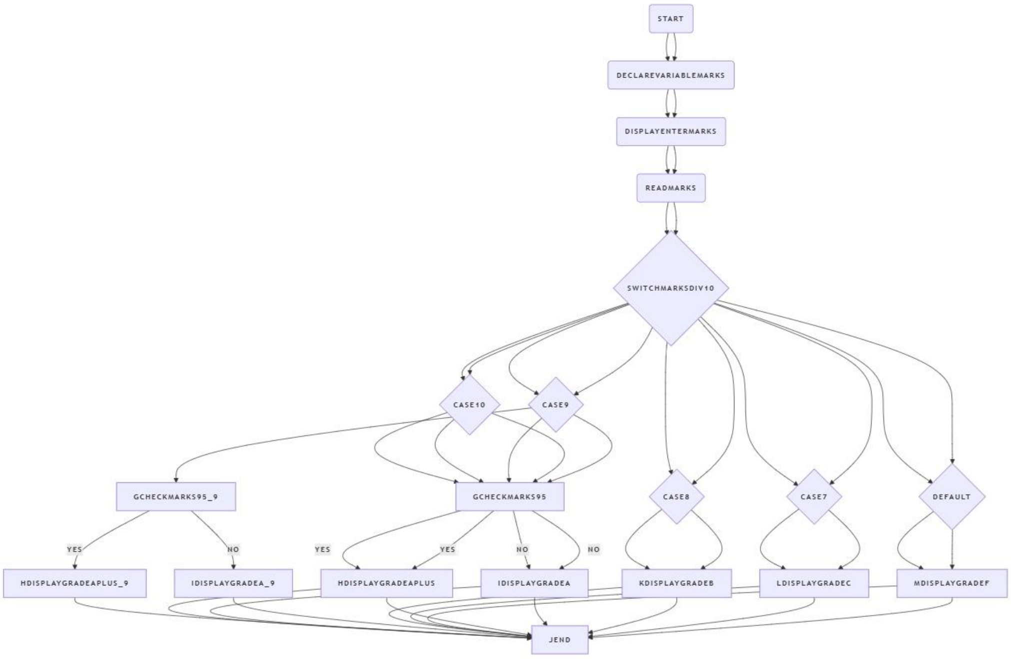
5.**Use switch(marks / 10)** to determine the grade category

6.**Check cases:**

- **Case 10:** (when marks = 100)
- **Case 9:** (when marks = 90–99)
  - If marks >= 95
    → **Display:** “Grade: A+”
  - Else
    → **Display:** “Grade: A”
  - **Break**
- **Case 8:** (when marks = 80–89)
  → **Display:** “Grade: B”
  → **Break**
- **Case 7:** (when marks = 70–79)
  → **Display:** “Grade: C”
  → **Break**
- **Default:** (when marks < 70)
  → **Display:** “Grade: F”

7.**End**

FLOWCHART



Q31. Print the multiplication table with a c program.

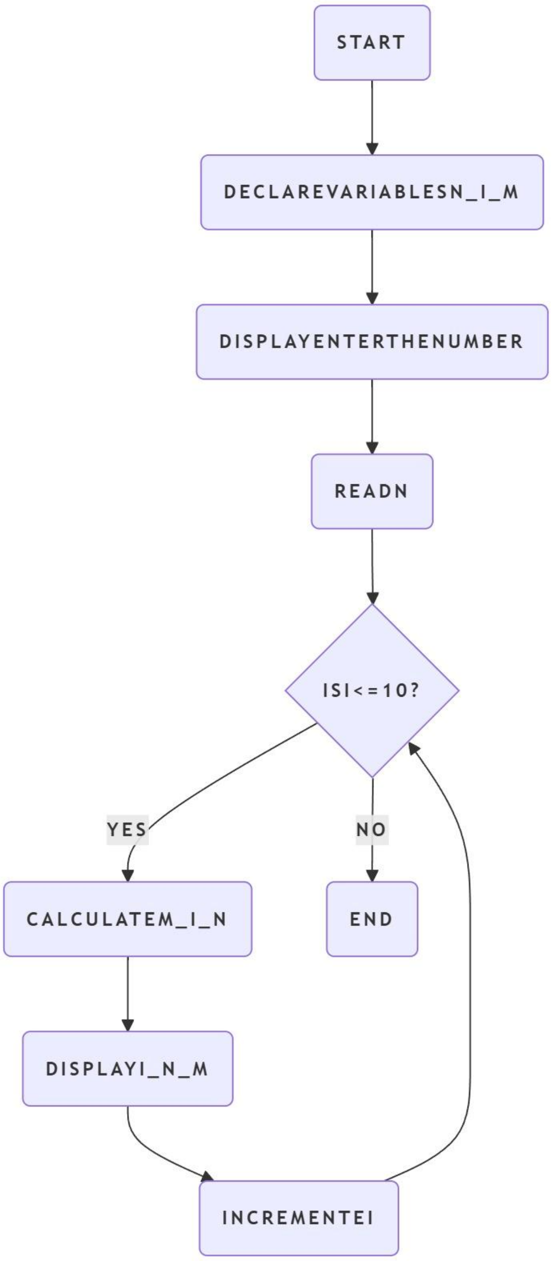
Ans:

ALGORITHM

1. **Start**
2. **Declare** integer variables: n, i, and m
3. **Display** message: “Enter the number”

4. **Read** the value of n
5. **Repeat** the following steps for i = 1 to 10:
    - Calculate m = i * n
    - **Display:** i * n = m
6. **End Loop**
7. **End**

FLOWCHART



Q32.Write a c program to find the sum of the n natural numbers.

Ans:

ALGORITHM

1. **Start**

2. **Declare** integer variables: n, i, and sum = 0

3. **Display** message: “Enter the range”

4. **Read** the value of n

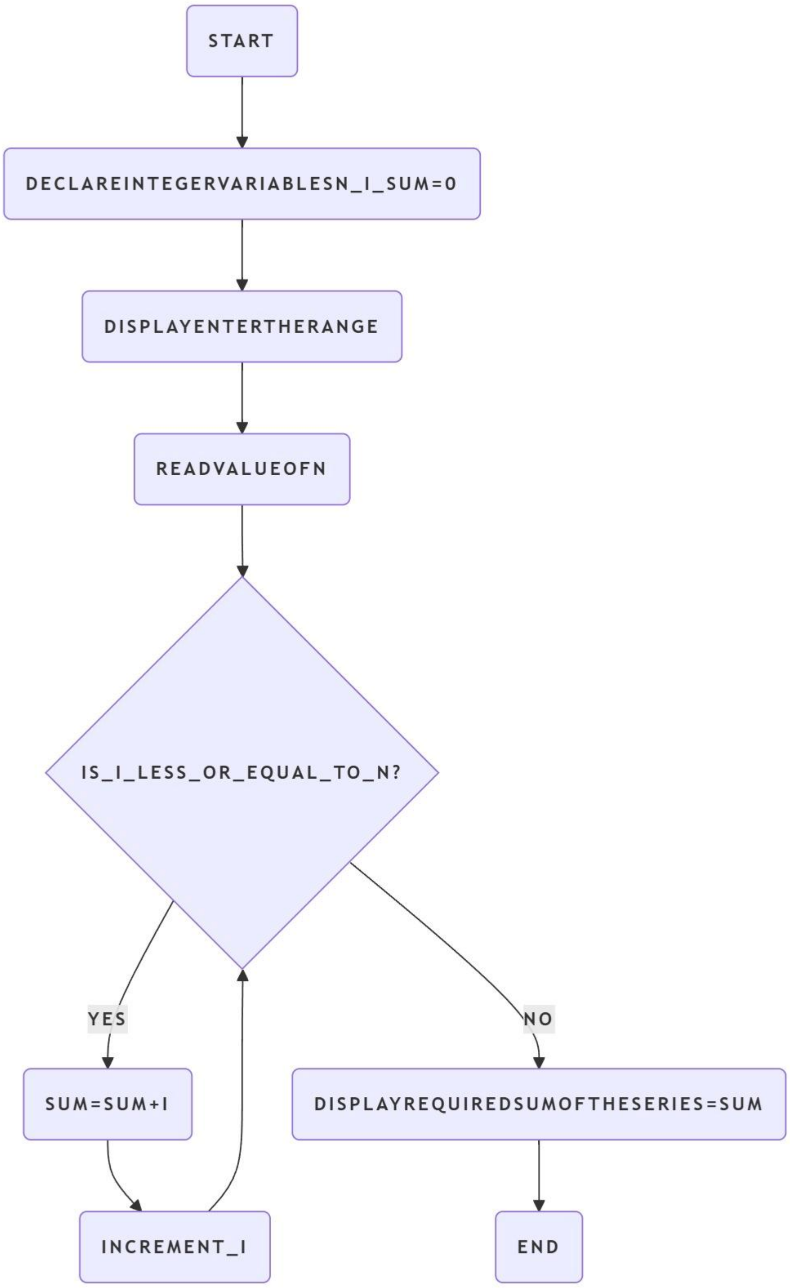
5.**Repeat** the following steps for i = 1 to n:

- Add i to sum → sum = sum + i

6. **After the loop ends**, **Display:** “Required sum of the series = sum”

7. **END**

FLOWCHART



Q33.Write a c program to print the factorial of a number using for loop.

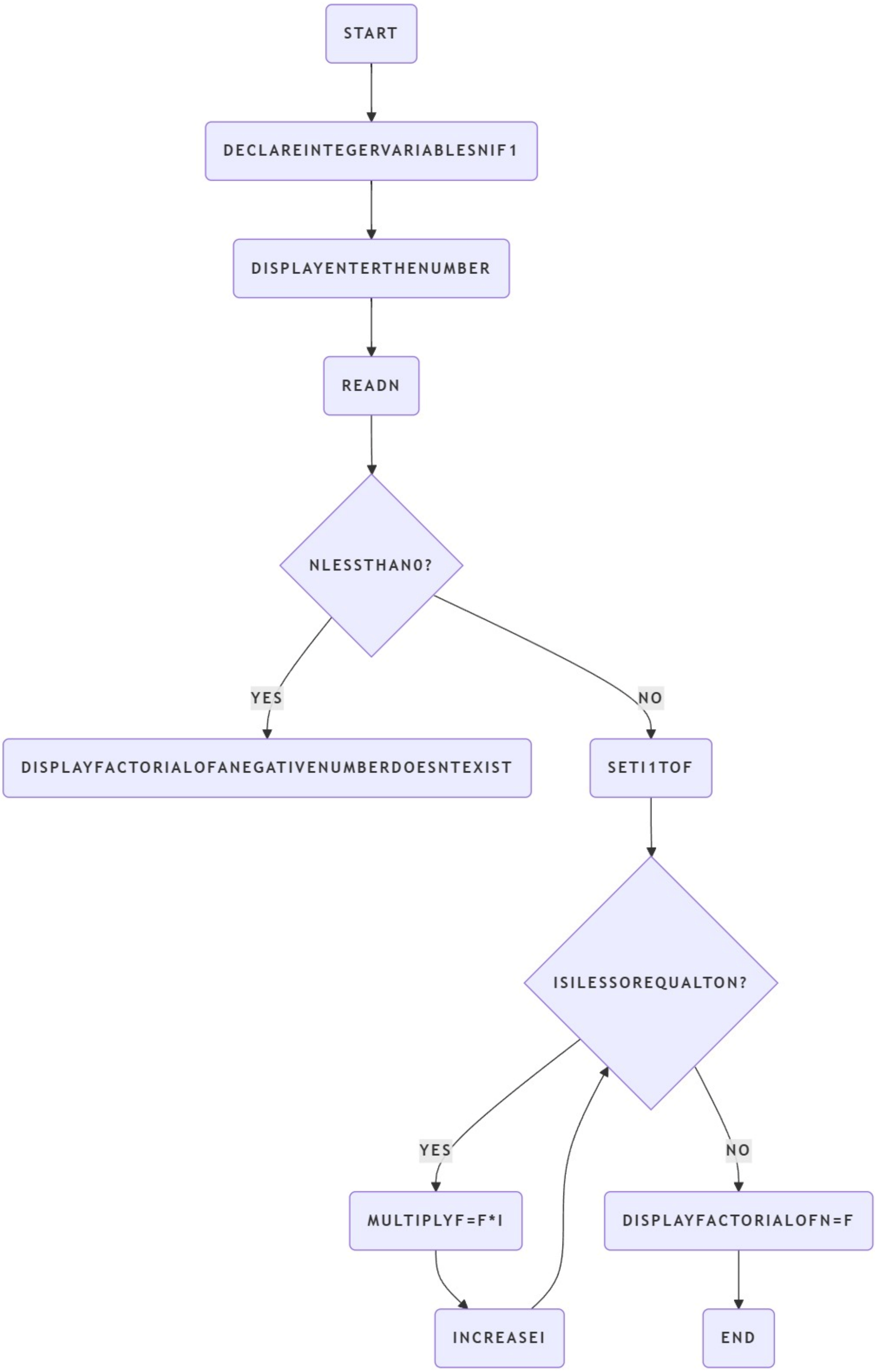
Ans:

ALGORITHM

1. **Start**
2. **Declare** integer variables: n, i, and f = 1
3. **Display** message: “Enter the number”

```
4. Read the value of n
5. If n < 0
      o  Display: “Factorial of a negative number doesn’t exist.”
6. Else
      o  Repeat the following steps for i = 1 to n:
            ▪  Multiply f = f * i
      o  Display: “Factorial of n = f”
7. End If
8. End
```

FLOWCHART



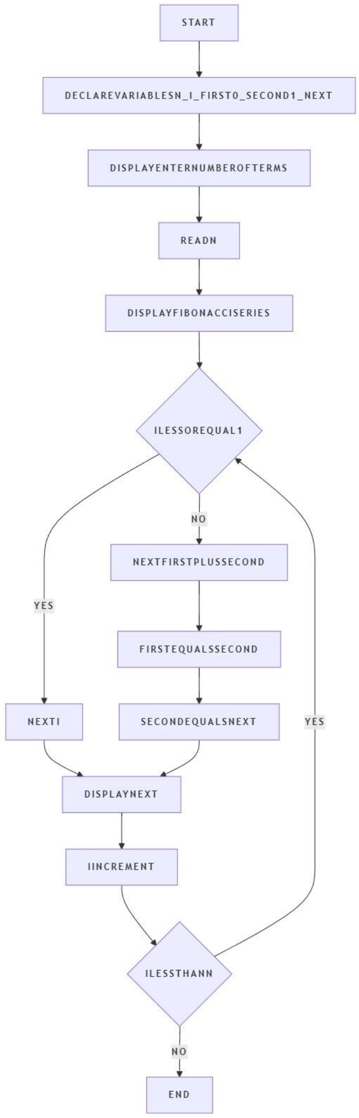
Q34.Write a c program to print Fibonacci series upto n terms.

Ans:

ALGORITHM

```
1. Start
2.Declare integer variables: n, i, first = 0, second = 1, next
3.Display message: “Enter the number of terms”
4.Read the value of n
5.Display message: “Fibonacci Series:”
6.Repeat the following steps for i = 0 to n - 1:
    • If i <= 1
      → next = i
    • Else
      → next = first + second
      → first = second
      → second = next
    • Display the value of next
7. End Loop
8.End
```

FLOWCHART



Q35. Write a c program to print all the prime numbers in a series.

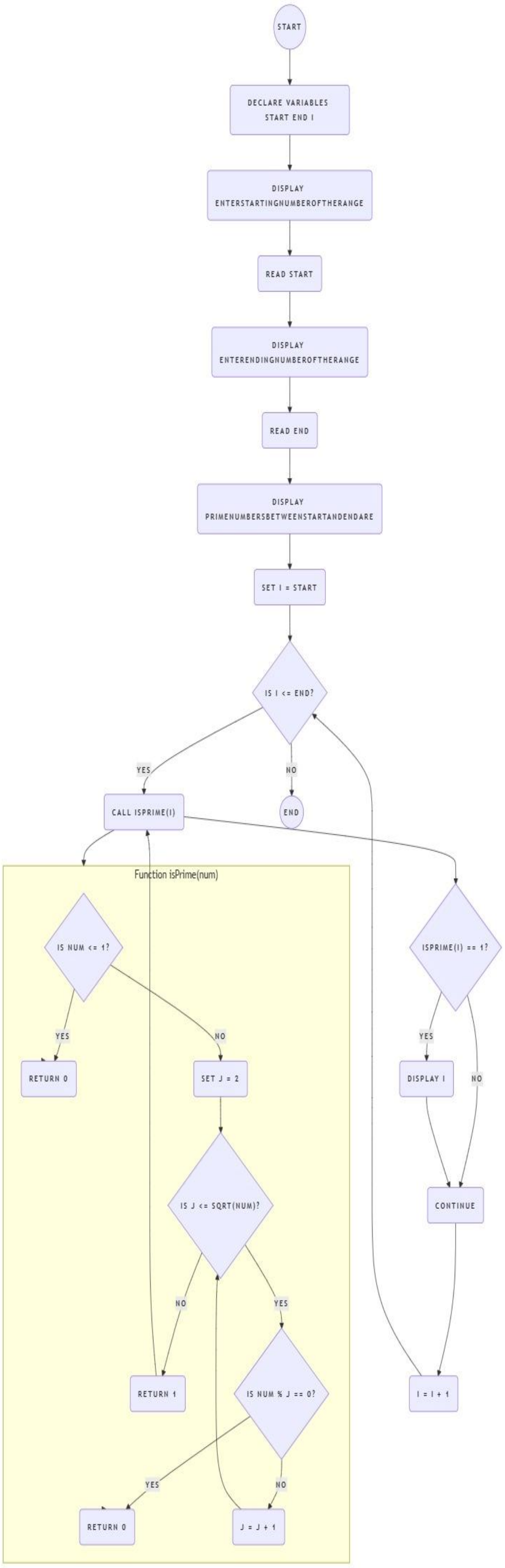
Ans:

ALGORITHM

1. **Start**
2. **Declare** integer variables: start, end, and i
3. **Display** message: “Enter the starting number of the range”
4. **Read** the value of start
5. **Display** message: “Enter the ending number of the range”
6. **Read** the value of end
7. **Display:** “Prime numbers between start and end are:”

8. **Repeat** for each number i from start to end:
    - **Call function** isPrime(i)
        - **Inside isPrime(num):**
            1. If num <= 1, **return 0** (not prime)
            2. For each integer j from 2 to √num:
                - If num % j == 0, **return 0** (not prime)
            3. **Return 1** (prime)
    - **If** isPrime(i) returns 1
      → **Display** i
9. **End Loop**
10. **End**

FLOWCHART


START
DECLARE VARIABLES
START END I
DISPLAY
ENTERSTARTINGNUMBEROFTHERANGE
READ START
DISPLAY
ENTERENDINGNUMBEROFTHERANGE
READ END
DISPLAY
PRIMENUMBERSBETWEENSTARTANDENDARE
SET I = START
IS I <= END?
YES
NO
CALL ISPRIME(I)
END
Function isPrime(num)
IS NUM <= 1?
YES
NO
RETURN 0
SET J = 2
IS J <= SQRT(NUM)?
NO
YES
RETURN 1
IS NUM % J == 0?
YES
NO
RETURN 0
J = J + 1
ISPRIME(I) == 1?
YES
NO
DISPLAY I
CONTINUE
I = I + 1