

# Contents

<b>CHAPTER - 1: GETTING STARTED</b>	<b>2</b>
PROBLEM 1.1 . . . . .	2
ALGORITHM . . . . .	2
PSEUDOCODE . . . . .	3
FLOWCHART . . . . .	4
PROBLEM 1.2 . . . . .	4
ALGORITHM: . . . . .	5
PSEUDOCODE: . . . . .	5
FLOWCHART: . . . . .	6
PROBLEM 1.3 . . . . .	6
ALGORITHM: . . . . .	6
PSEUDOCODE: . . . . .	7
FLOWCHART . . . . .	8
<b>CHAPTER-2: C INSTRUCTIONS</b>	<b>9</b>
PROBLEM 2.1 . . . . .	9
ALGORITHM . . . . .	9
PSEUDOCODE . . . . .	9
FLOWCHART . . . . .	10
PROBLEM 2.2 . . . . .	11
ALGORITHM . . . . .	11
PSEUDOCODE . . . . .	11
FLOWCHART . . . . .	12
PROBLEM 2.3 . . . . .	13
ALGORITHM: . . . . .	13
PSEUDOCODE . . . . .	13
FLOWCHART . . . . .	14
PROBLEM 2.4 . . . . .	15
ALGORITHM . . . . .	15
PSEUDOCODE . . . . .	15
FLOWCHART . . . . .	16
PROBLEM 2.5 . . . . .	17
ALGORITHM . . . . .	17
PSEUDOCODE . . . . .	17
FLOWCHART . . . . .	18
PROBLEM 2.6 . . . . .	19
ALGORITHM . . . . .	19
PSEUDOCODE . . . . .	19
FLOWCHART . . . . .	20
<b>CHAPTER - 3: CONDITIONAL STATEMENTS</b>	<b>21</b>
PROBLEM 3.1 . . . . .	21
ALGORITHM . . . . .	21
PSEUDOCODE . . . . .	21

FLOWCHART . . . . .	23
PROBLEM 3.2 . . . . .	24
ALGORITHM . . . . .	24
PSEUDOCODE . . . . .	24
FLOWCHART . . . . .	25
PROBLEM 3.3 . . . . .	26
ALGORITHM . . . . .	26
PSEUDOCODE: . . . . .	26
FLOWCHART . . . . .	27
PROBLEM 3.4 . . . . .	27
ALGORITHM . . . . .	27
PSEUDOCODE . . . . .	28
FLOWCHART . . . . .	29
PROBLEM 3.5 . . . . .	29
ALGORITHM . . . . .	30
PSEUDOCODE . . . . .	30
FLOWCHART . . . . .	31
PROBLEM 3.6 . . . . .	31
ALGORITHM . . . . .	31
PSEUDOCODE . . . . .	32
FLOWCHART . . . . .	33
PROBLEM 3.7 . . . . .	34
ALGORITHM . . . . .	34
PSEUDOCODE . . . . .	34
FLOWCHART . . . . .	35
PROBLEM 3.8 . . . . .	35
ALGORITHM . . . . .	35
PSEUDOCODE . . . . .	36
FLOWCHART . . . . .	36
PROBLEM 3.9 . . . . .	37
ALGORITHM . . . . .	37
PSEUDOCODE . . . . .	37
FLOWCHART . . . . .	39

## CHAPTER - 1: GETTING STARTED

### PROBLEM 1.1

Temperature of a city in Farenheit degrees is input through the keyboard. Write a program to convert this temperature into Centigrade degrees.

#### ALGORITHM

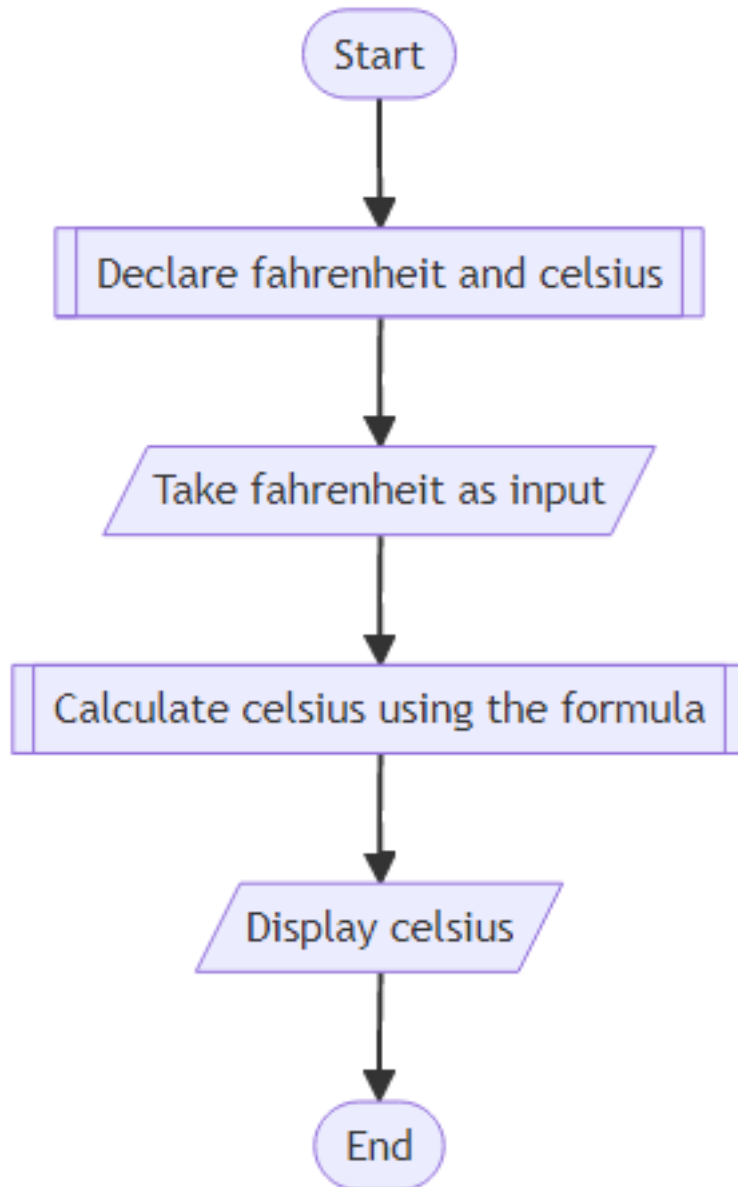
1. Start
2. Declare float variables Farenheit and celsius
3. Take farenheit as input

4. Calculate Celsius from the formula  $celsius = ((fahrenheit - 32) / 1.8)$
5. Display the fahrenheit and celsius values
6. Stop

#### **PSEUDOCODE**

```
DECLARE FLOAT fahrenheit, celsius
INPUT fahrenheit
ASSIGN celsius to ((fahrenheit - 32) / 1.8);
DISPLAY "Value in fahrenheit is: "
DISPLAY fahrenheit
DISPLAY "Value in celsius is: "
DISPLAY celsius
```

## FLOWCHART



## PROBLEM 1.2

The length and breadth of a rectangle and radius of a circle are input through the keyboard. Write a program to calculate the area and perimeter of the rectangle,

and the area and circumference of the circle.

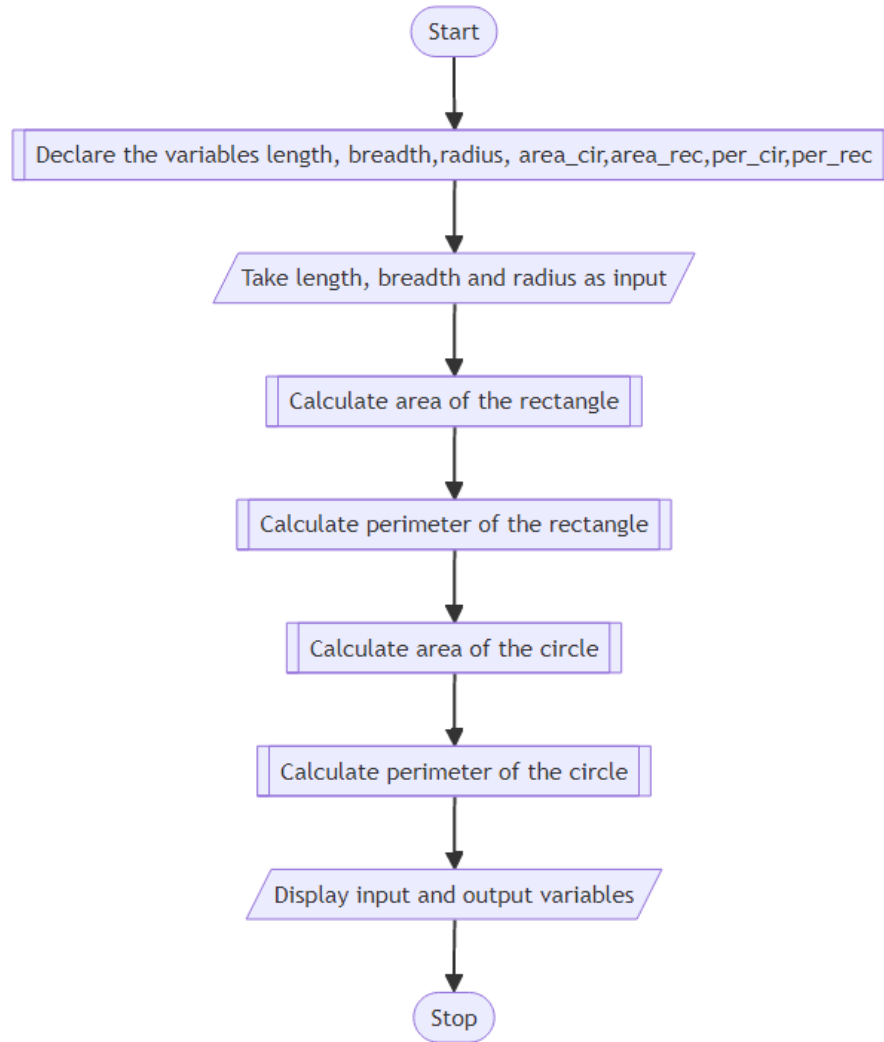
**ALGORITHM:**

1. Start
2. Declare float variables length, breadth, radius, area\_cir, area\_rec, per\_cir, per\_rec
3. Take length, breadth and radius as input
4. Calculate the area of the rectangle using the formula  $\text{area\_rec} = \text{length} * \text{breadth}$ ;
5. Calculate the perimeter of the rectangle using the formula  $\text{per\_rec} = 2 * (\text{length} + \text{breadth})$ ;
6. Calculate the area of the circle using the formula  $\text{area\_cir} = 3.14 * \text{radius}^2$ ;
7. Calculate the perimeter of the circle using the formula  $\text{per\_cir} = 2 * 3.14 * \text{radius}$ ;
8. Display the input and output variables
9. Stop

**PSEUDOCODE:**

```
DECLARE FLOAT length, breadth, radius, area_cir, area_rec, per_cir, per_rec
INPUT length, breadth, radius
ASSIGN area_rec to length*breadth
ASSIGN per_rec to 2*(length+breadth)
ASSIGN area_cir to 3.14*radius^2
ASSIGN per_cir to 2*3.14*radius
DISPLAY "The length of the rectangle:"
DISPLAY length
DISPLAY "The breadth of the rectangle:"
DISPLAY breadth
DISPLAY "The radius of the circle:"
DISPLAY radius
DISPLAY "The area of the rectangle:"
DISPLAY area_rec
DISPLAY "The perimeter of the rectangle:"
DISPLAY per_rec
DISPLAY "The area of the circle:"
DISPLAY area_cir
DISPLAY "The perimeter of the circle:"
DISPLAY per_cir
```

### FLOWCHART:



### PROBLEM 1.3

Paper of size A0 has dimensions 1189 cm x 841 mm. Each subsequent size A(n) is defined as A(n-1) cut in half, parallel to its shorter sides. Thus, a paper of size A1 would have dimensions 841 mm x 594 mm. Write a program to calculate and print the paper sizes A1 to A8.

### ALGORITHM:

1. Start

2. Declare float variables h and w.
3. Assign h to 1189 and w to 841.
4. Display “size of A0: h x w”
5. Calculate the size of A1 as:  $w = 2^{(-0.25 - 0.5 * i)} * 1000$ ,  $h = 2^{(0.25 - 0.5 * i)} * 1000$ , taking i as 1.
6. Repeat Step 5 for i = 2 to i= 8
7. Display the sizes of A1 to A8.
8. Stop

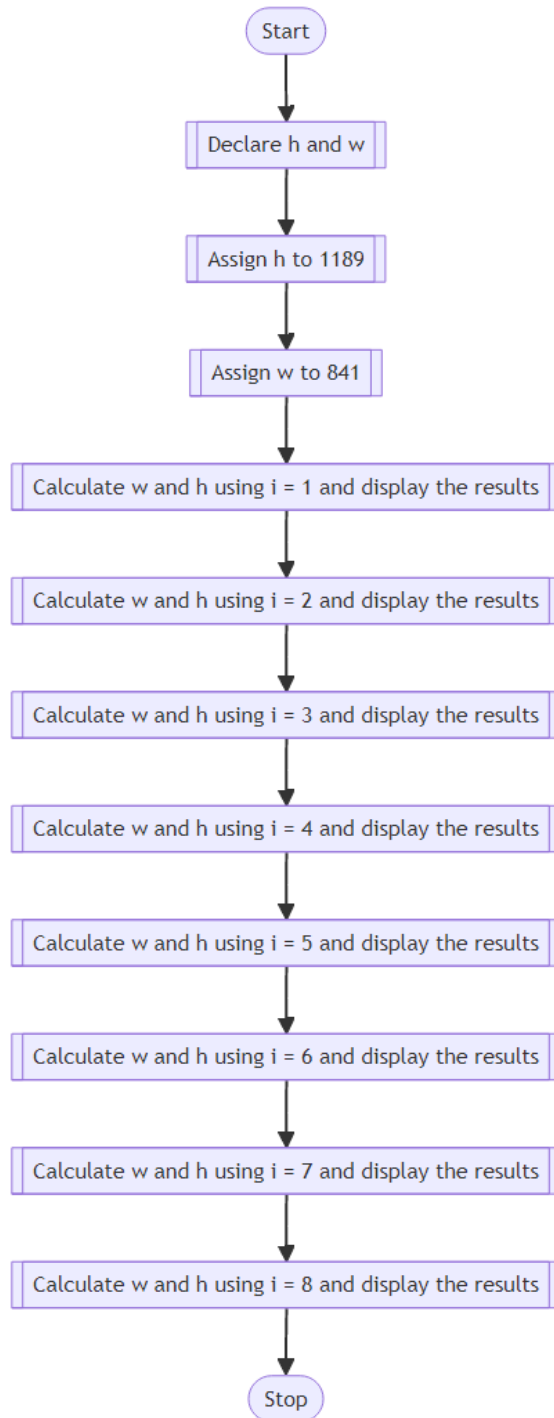
#### PSEUDOCODE:

```

DECLARE FLOAT h,w
ASSIGN h to 1189
ASSIGN w to 841
DECLARE INTEGER i
ASSIGN i to 0
DISPLAY "Size of A0: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 1) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 1) * 1000
DISPLAY "Size of A1: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 2) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 2) * 1000
DISPLAY "Size of A2: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 3) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 3) * 1000
DISPLAY "Size of A3: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 4) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 4) * 1000
DISPLAY "Size of A4: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 5) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 5) * 1000
DISPLAY "Size of A5: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 6) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 6) * 1000
DISPLAY "Size of A6: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 7) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 7) * 1000
DISPLAY "Size of A7: h x w"
ASSIGN w to 2^(-0.25 - 0.5 * 8) * 1000
ASSIGN h to 2^(0.25 - 0.5 * 8) * 1000
DISPLAY "Size of A8: h x w"

```

## FLOWCHART





## CHAPTER-2: C INSTRUCTIONS

### PROBLEM 2.1

If a five-digit number is input through the keyboard, write a program to calculate the sum of its digits.

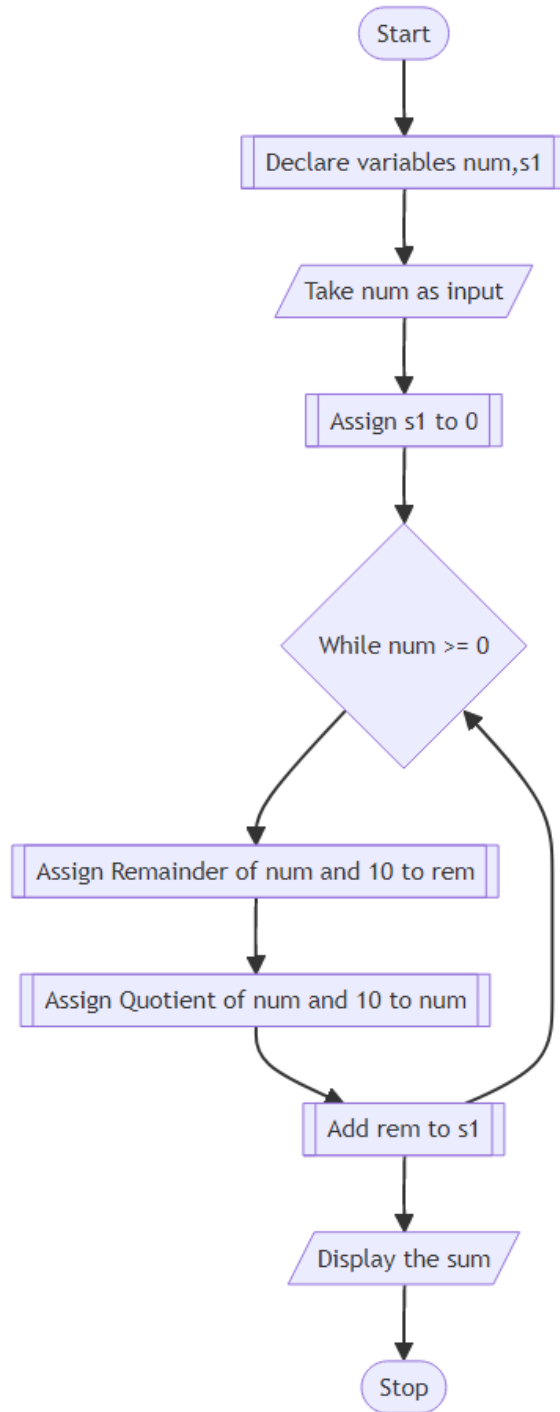
#### ALGORITHM

1. Start
2. Declare int variables num, and s1
3. Take num as input
4. Assign s1 to 0
5. divide num by 10 and store the remainder in variable rem.
6. add rem to sum.
7. divide num by 10 and re-assign num to be the quotient.
8. Repeat steps 5-7 until num becomes less than or equal to zero
9. display the sum s1.
10. Stop

#### PSEUDOCODE

```
DECLARE INTEGER num,s1
INPUT num
ASSIGN s1 to 0
WHILE num >= 0
    ASSIGN Remainder(num,10) to rem
    ASSIGN num/10 to num
    ADD rem to s1
ENDWHILE
DISPLAY "The sum of digits of number is: "
DISPLAY s1
```

## FLOWCHART



## PROBLEM 2.2

Write a program to receive Cartesian Coordinates (x,y) of a point and convert them into polar coordinates (r,phi) **Hint:  $r = \sqrt{x^2+y^2}$ ,  $\phi = \tan^{-1}(y/x)$**

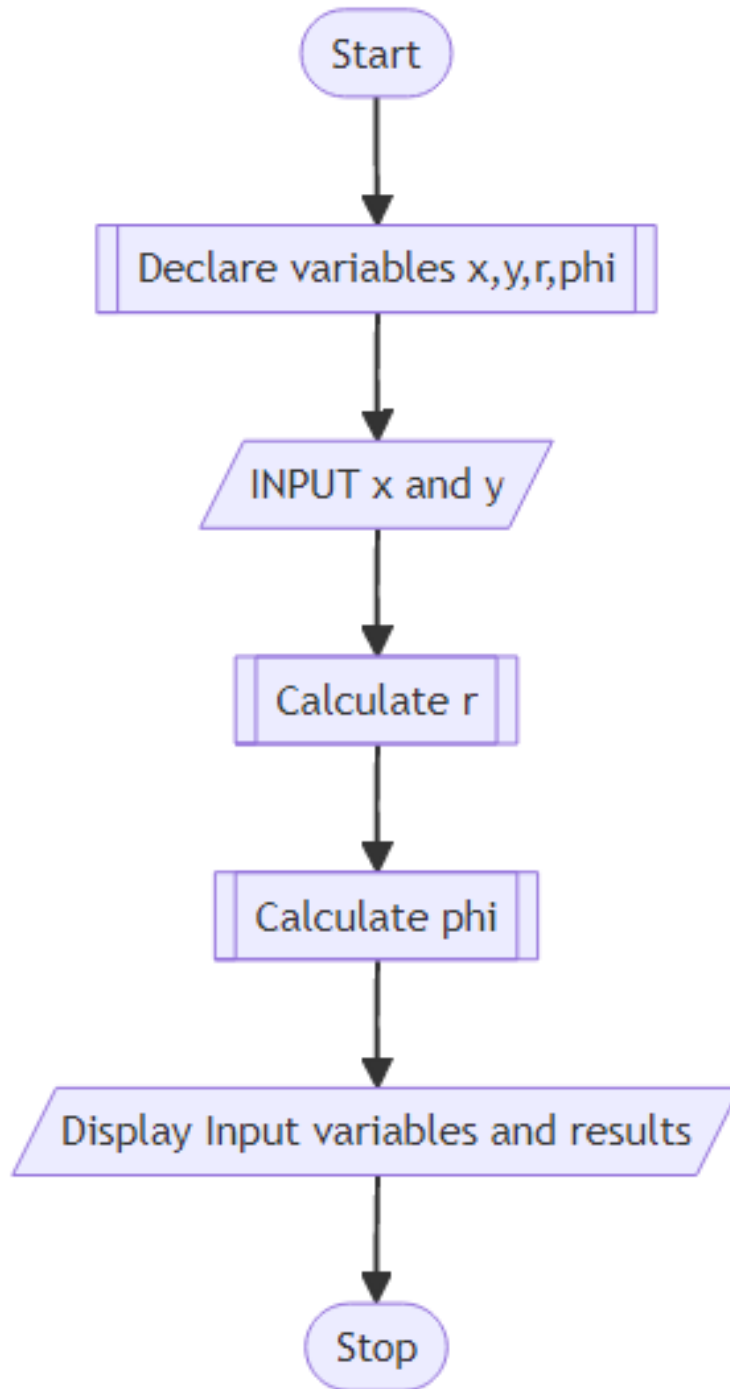
### ALGORITHM

1. Start
2. Declare float variables x,y,r,phi
3. Take x and y as input
4. Calculate r using the formula  $r = \sqrt{x^2+y^2}$
5. Calculate phi using the formula  $\phi = \tan^{-1}(y/x)$
6. Display the input variables and results
7. Stop

### PSEUDOCODE

```
DECLARE FLOAT x,y,r,phi
INPUT x,y
ASSIGN r to SQUARE ROOT (x square + y square)
ASSIGN phi to INVERSE TANGENT(y/x)
DISPLAY "Cartesian Co-ordinates: "
DISPLAY x and y
DISPLAY "Polar Coordinates: "
DISPLAY r and phi
```

## FLOWCHART



### PROBLEM 2.3

Write a program to receive values of latitude(L1,L2) and longitude(G1,G2), in degrees, of two places on earth and output the distance between the distance(D) between them in nautical miles. The formula for the distance in nautical miles is:  $D = 3963 \cos^{-1}(\sin L1 \sin L2 + \cos L1 \cos L2 \cos(G2-G1))$

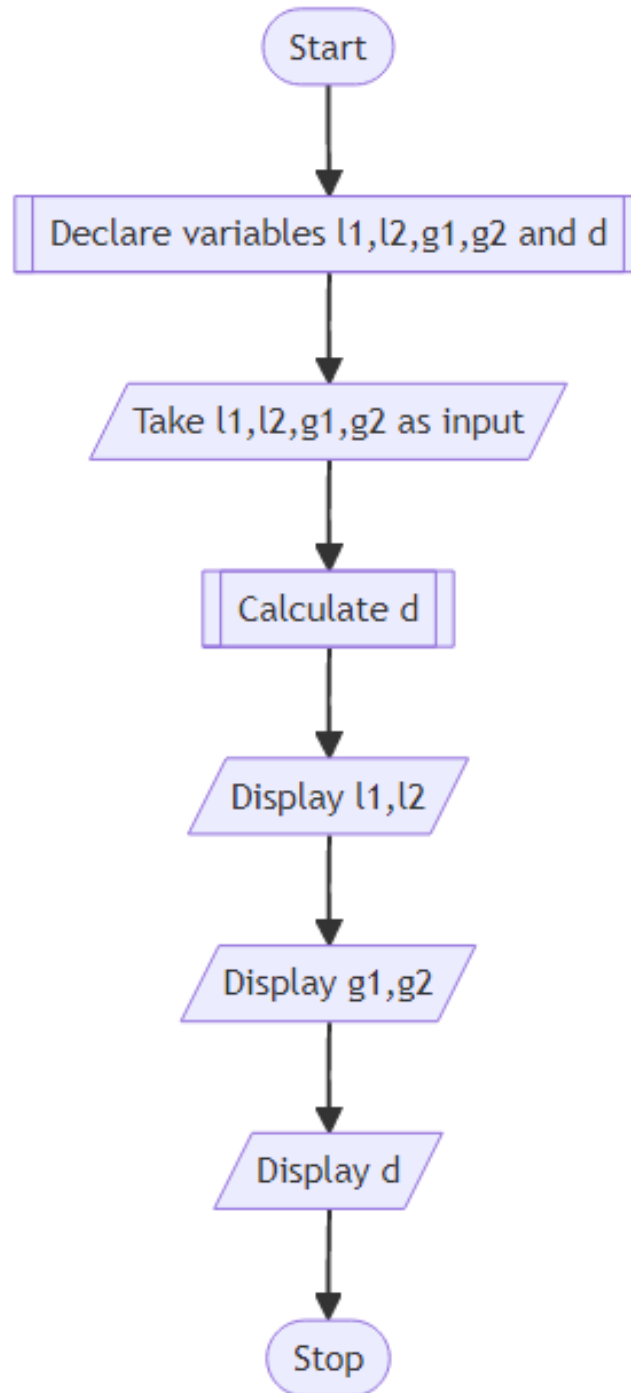
#### ALGORITHM:

1. Start
2. Declare float variables l1,l2,g1,g2 and d
3. Take l1,l2,g1 and g2 as input
4. Calculate d using  $d = 3963 \cos^{-1}(\sin L1 \sin L2 + \cos L1 \cos L2 \cos(G2-G1))$
5. Display the input and results
6. Stop

#### PSEUDOCODE

```
DECLARE FLOAT l1,l2,g1,g2,d
INPUT l1,l2,g1,g2
ASSIGN d to 3963 cos-1(sin L1 sin L2+ cos L1 cos L2*cos(G2-G1))
DISPLAY "Latitudes: "
DISPLAY l1,l2
DISPLAY "Longitudes: "
DISPLAY g1,g2
DISPLAY "Distance in nautical miles: "
DISPLAY d
```

## FLOWCHART



## PROBLEM 2.4

Wind chill factor is the felt air temperature on exposed skin due to wind. The wind chill temperature is always lower than the air temperature, and is calculated using the following formula:  $wcf = 35.74 + 0.6215t + (0.4275t - 35.75)v^{0.16}$  where  $t$  is temperature,  $v$  is wind velocity. Write a program to receive values of  $t$  and  $v$  and calculate  $wcf$ .

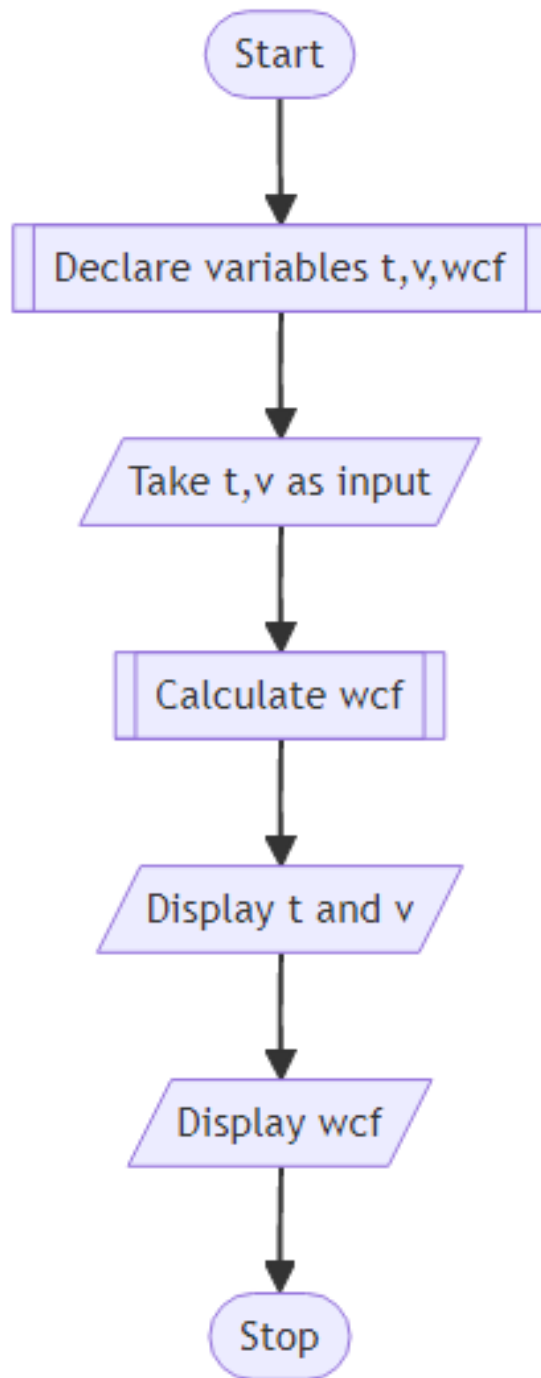
## ALGORITHM

1. Start
2. Declare float variables  $t, v$  and  $wcf$
3. Take  $t$  and  $v$  as input
4. Calculate  $wcf$  using:  $wcf = 35.74 + 0.6215t + (0.4275t - 35.75)v^{0.16}$
5. Display the inputs and results.
6. Stop

## PSEUDOCODE

```
DECLARE FLOAT t,v,wcf
INPUT t,v
ASSIGN wcf to 35.74+0.6215t +(0.4275t - 35.75)*v^0.16
DISPLAY "Temperature : "
DISPLAY t
DISPLAY "Velocity : "
DISPLAY v
DISPLAY "Wind chill factor : "
DISPLAY wcf
```

## FLOWCHART





## PROBLEM 2.5

If value of an angle is input through the keyboard, write a program to print all its trigonometric ratios.

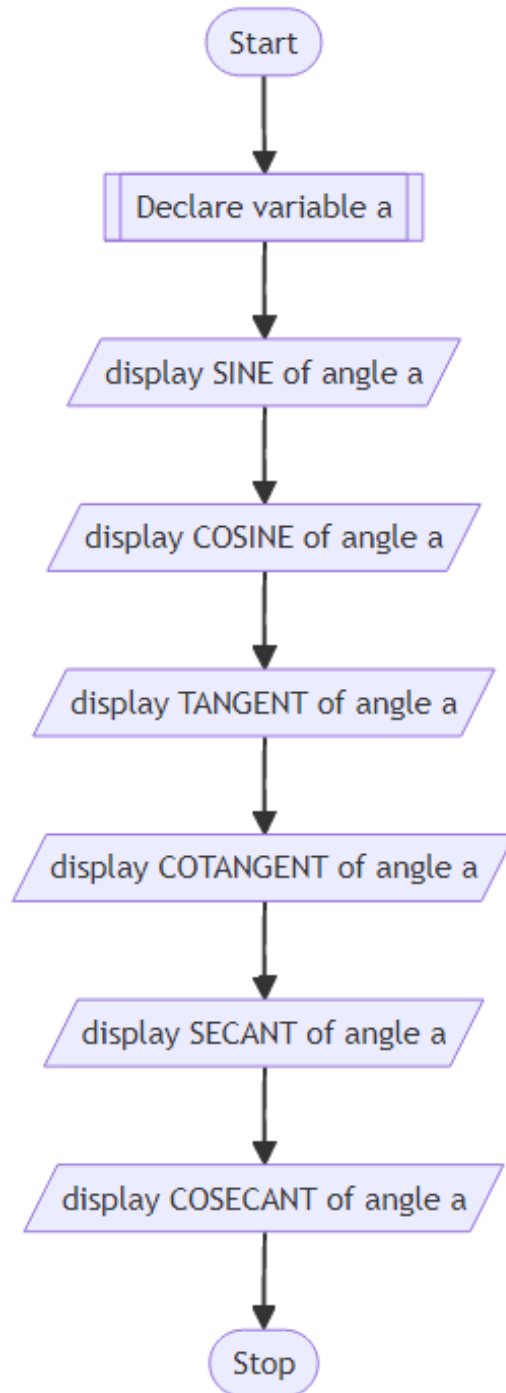
### ALGORITHM

1. Start
2. Declare float variable a
3. Calculate  $\sin(a)$  and print the result
4. Calculate  $\cos(a)$  and print the result
5. Calculate  $\tan(a)$  and print the result
6. Calculate  $\cot(a)$  and print the result
7. Calculate  $\sec(a)$  and print the result
8. Calculate  $\operatorname{cosec}(a)$  and print the result
9. Stop

### PSEUDOCODE

```
DECLARE FLOAT a
DISPLAY "SINE of angle a is : sin(a)"
DISPLAY "COSINE of angle a is : cos(a)"
DISPLAY "TANGENT of angle a is : tan(a)"
DISPLAY "COTANGENT of angle a is : cot(a)"
DISPLAY "SECANT of angle a is : sec(a)"
DISPLAY "COSECANT of angle a is : cosec(a)"
```

## FLOWCHART



## PROBLEM 2.6

Two numbers are input through two locations c and d. Write a program to interchange contents of c and d.

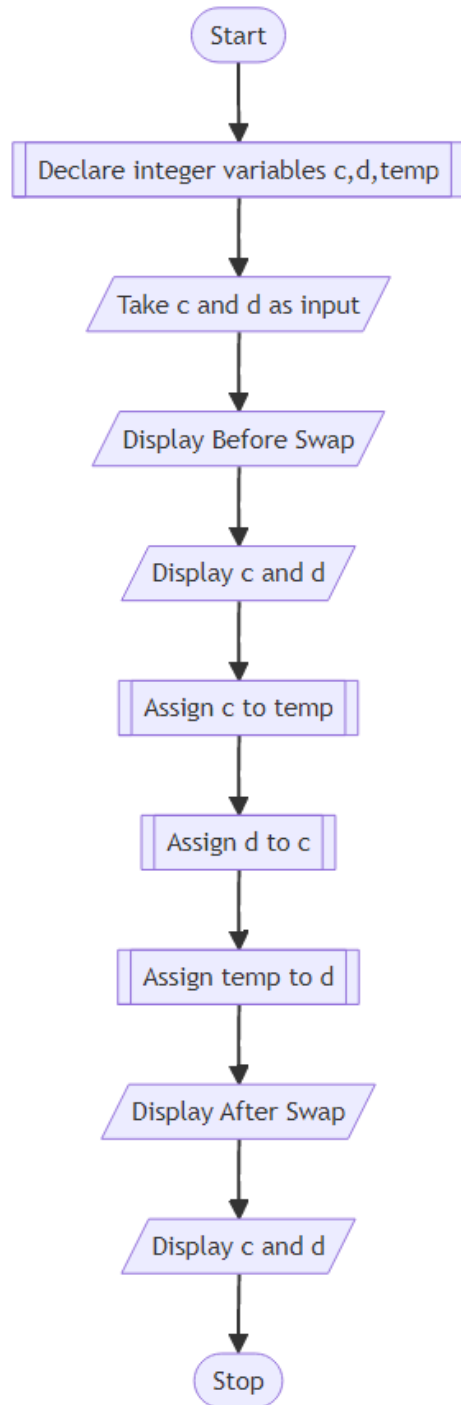
### ALGORITHM

1. Start
2. Declare integer variables a and b
3. Declare another variable temp
4. Take a and b as input
5. Display the values of the variables before swap
6. Assign a to temp
7. Assign b to a
8. Assign temp to b
9. Display values of a and b'
10. Stop

### PSEUDOCODE

```
DECLARE INTEGER a and b
DECLARE INTEGER temp
INPUT a,b
DISPLAY "Before Swap"
DISPLAY a,b
ASSIGN a to temp
ASSIGN b to a
ASSIGN temp to b
DISPLAY "After Swap"
DISPLAY a,b
```

## FLOWCHART



## CHAPTER - 3: CONDITIONAL STATEMENTS

### PROBLEM 3.1

A five-digit number is entered through the keyboard. Write a program to obtain the reversed number and to determine whether the original and reversed numbers are equal or not

#### ALGORITHM

1. Start
2. Declare integer variable num, and an array a
3. Take num as input
4. Take the remainder of num and 10 and add to the array.
5. Reassign num to integer division of num and 10
6. Repeat steps 4 and 5 until num $\leq$ 0
7. Declare a variable s
8. Multiply  $2^{*(\text{length}(a)-i)}$  to the ith element of a, and add to the variable s
9. Repeat step 8 for each element in a
10. check if s is equal to num:
  - a. If yes, display "Pallindrome"
  - b. Else, display "Not pallindrome"
11. Stop

#### PSEUDOCODE

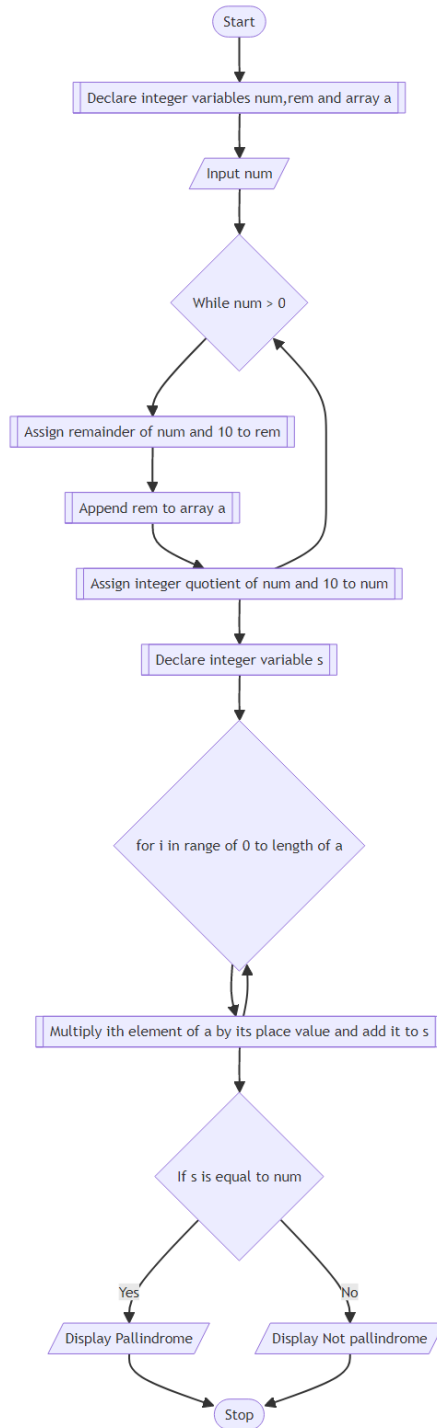
```
FUNCTION CountDigits(INTEGER num)
    DECLARE INTEGER result and ASSIGN 0 to it
    WHILE num != 0
        ASSIGN num/10 to num
        INCREMENT result
    ENDWHILE
    RETURN result
ENDFUNCTION

FUNCTION main
    DECLARE INTEGER num,rem,temp,n
    INPUT num
    ASSIGN CountDigits(num) to n
    DECLARE INTEGER ARRAY a[n]
    ASSIGN num to temp
    WHILE temp>0
        ASSIGN REMAINDER(temp,10) to rem
        APPEND rem to ARRAY a
        ASSIGN temp/10 to temp
```

```
ENDWHILE
DECLARE INTEGER s = 0
FOR i IN RANGE OF 0 to length(a)
    ASSIGN s to s*10 + a[i]
ENDFOR

IF s is equal to num
    DISPLAY "Pallindrome"
ELSE
    DISPLAY "Not Pallindrome"
ENDIF
ENDFUNCTION
```

## FLOWCHART



### PROBLEM 3.2

If ages of Ram, Shyam and Ajay are input through the keyboard, write a program to determine the youngest of the three

#### ALGORITHM

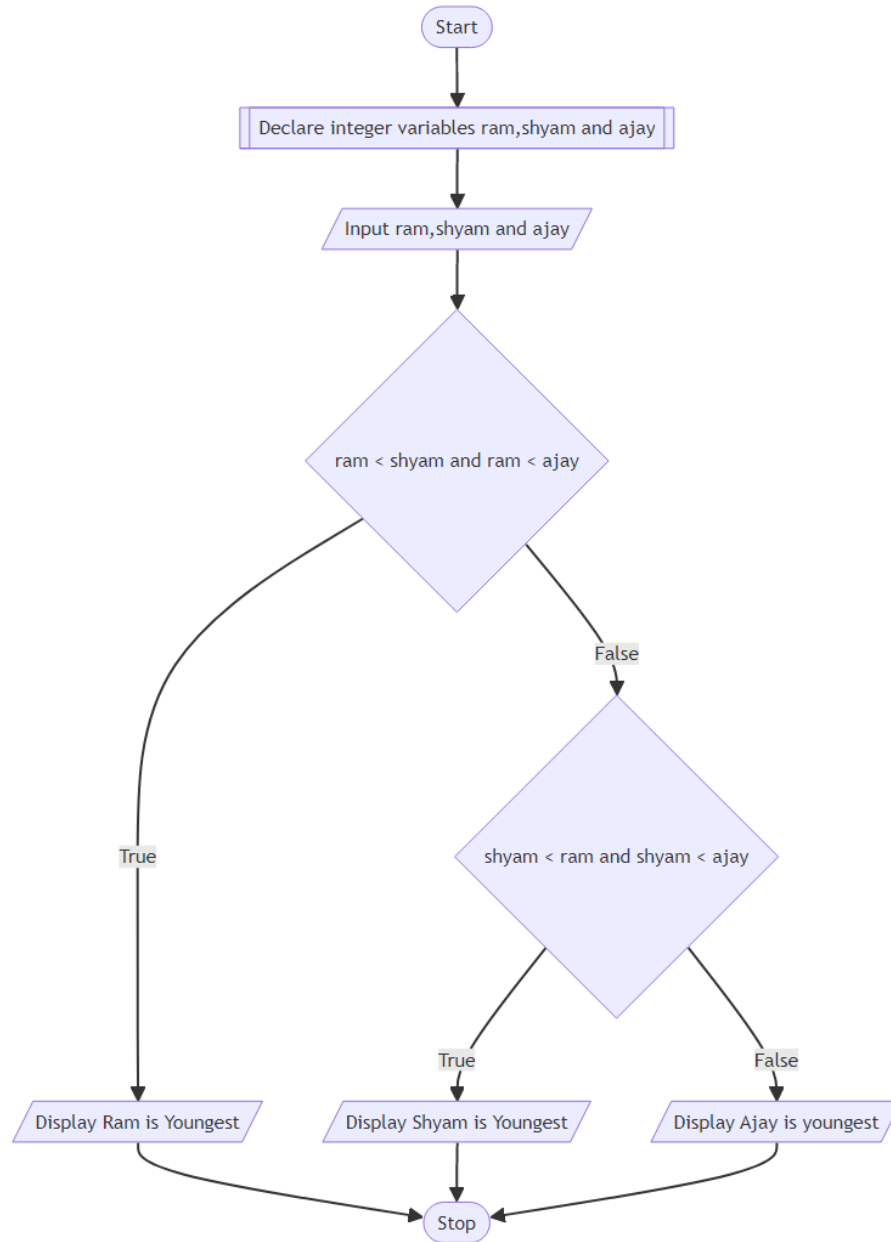
1. Start
2. Declare integer variables ram, shyam and ajay
3. Input ram, shyam and ajay
4. If ram is less than both shyam and ajay, display “ram is youngest”.
5. Else if shyam is less than both ram and ajay, display “shyam is youngest”.
6. Else display “Ajay is youngest”
7. Stop

#### PSEUDOCODE

```
DECLARE INTEGER ram,shyam,ajay
INPUT ram,shyam,ajay
IF ram < shyam and ram < ajay
    DISPLAY "Ram is youngest"
ELSE IF shyam < ram and shyam < ajay
    DISPLAY "Shyam is youngest"
ELSE
    DISPLAY "Ajay is youngest"
ENDIF
```



## FLOWCHART



### PROBLEM 3.3

Write a program to check whether a triangle is valid or not, if three angles of the triangle are entered through the keyboard. A triangle is valid if the sum of all the three angles is equal to 180 degrees

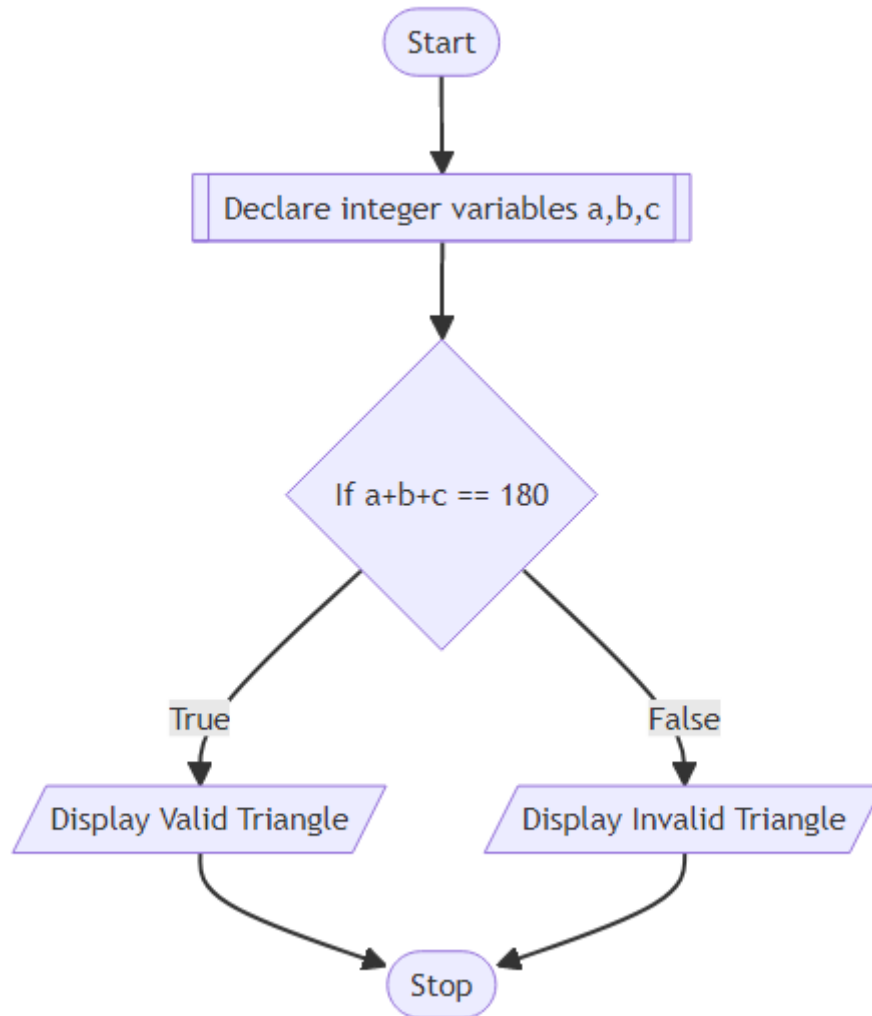
#### ALGORITHM

1. Start
2. Declare integer variables a,b,c
3. If  $a+b+c == 180$ , display "Valid Triangle"
4. Else, display "Invalid triangle"
5. Stop

#### PSEUDOCODE:

```
DECLARE INTEGER a,b,c
INPUT a,b,c
IF a+b+c == 180
    DISPLAY "Valid Triangle"
ELSE
    DISPLAY "Invalid Triangle"
ENDIF
```

## FLOWCHART



## PROBLEM 3.4

Write a program to find the absolute value of a number entered through the keyboard

## ALGORITHM

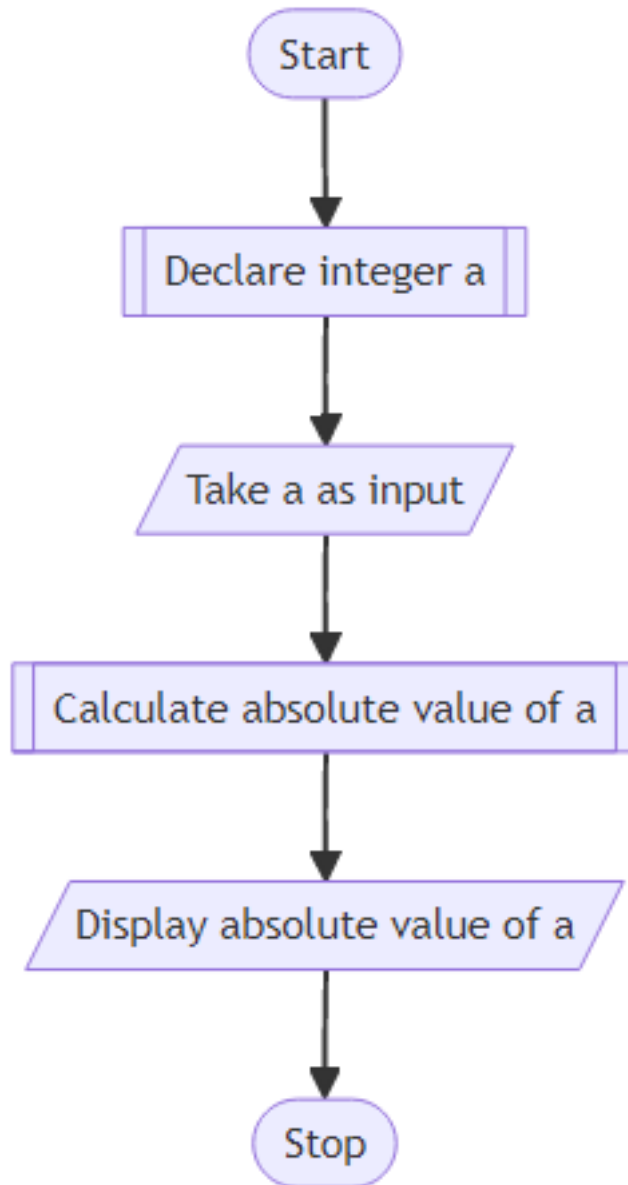
1. Start
2. Declare variable a
3. Input a

4. Calculate `abs(a)`
5. Display "Absolute value:",`abs(a)`
6. Stop

#### **PSEUDOCODE**

```
DECLARE INTEGER a
INPUT a
DISPLAY "Absolute value of a: "
DISPLAY abs(a)
```

## FLOWCHART



## PROBLEM 3.5

Given the length and breadth of a rectangle, write a program to find whether the area of the rectangle greater than its perimeter. For example, the area of

the rectangle with length = 5 and breadth = 4 is greater than its perimeter

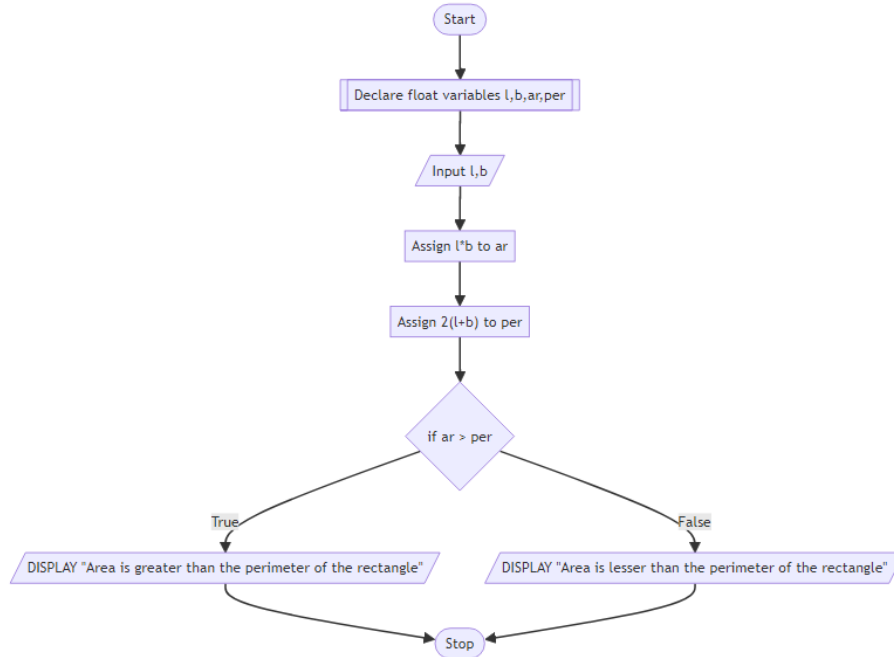
### ALGORITHM

1. Start
2. Declare float variables l,b,ar,per
3. Assign  $l*b$  to ar
4. Assign  $2(l+b)$  to per
5. If  $ar > per$ , display "Area is greater than the perimeter of the rectangle"
6. Else, display "Area is lesser than the perimeter of the rectangle"
7. Stop

### PSEUDOCODE

```
DECLARE FLOAT l,b,ar,per
INPUT l,b
ASSIGN l*b to ar
ASSIGN 2(l+b) to per
IF ar > per
    DISPLAY "Area is greater than the perimeter of the rectangle"
ELSE
    DISPLAY "Area is lesser than the perimeter of the rectangle"
ENDIF
```

## FLOWCHART



## PROBLEM 3.6

Given the three points  $X(x_1, y_1)$ ,  $Y(x_2, y_2)$  and  $Z(x_3, y_3)$ , write a program to check if the three points fall on straight line.

Condition for collinearity:

$$Slope(XY) = Slope(YZ) = Slope(XZ)$$

WHERE,

$$Slope(XY) = \frac{y_2 - y_1}{x_2 - x_1}$$

$$Slope(YZ) = \frac{y_3 - y_2}{x_3 - x_2}$$

$$Slope(XZ) = \frac{y_3 - y_1}{x_3 - x_1}$$

## ALGORITHM

1. Start
2. Declare float variables  $x_1, y_1, x_2, y_2, x_3, y_3, slope\_xy, slope\_yz, slope\_xz$

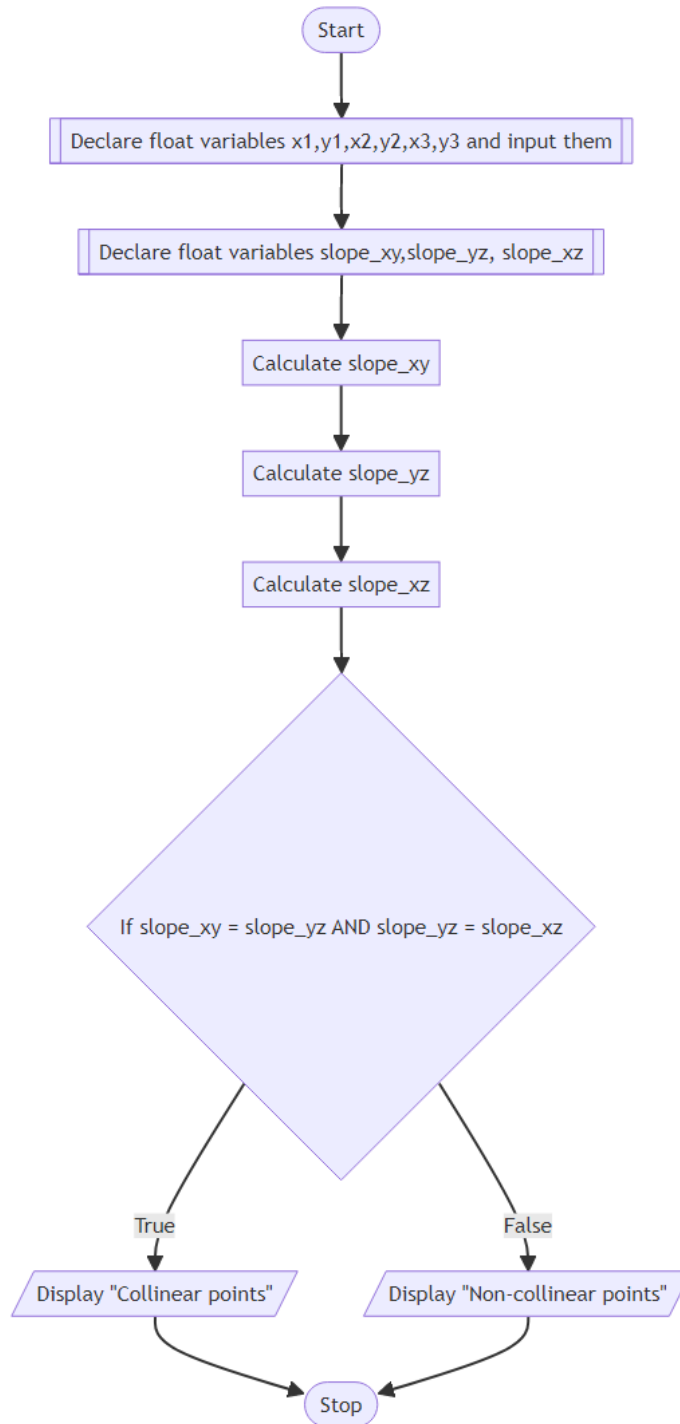
3. Input  $x_1, y_1, x_2, y_2, x_3, y_3$
4. Assign  $(y_2 - y_1) / (x_2 - x_1)$  to `slope_xy`
5. Assign  $(y_3 - y_2) / (x_3 - x_2)$  to `slope_yz`
6. Assign  $(y_3 - y_1) / (x_3 - x_1)$  to `slope_xz`
7. If `slope_xy = slope_yz` and `slope_yz = slope_xz`, Display “Collinear points”
8. Else, display “non-collinear points”
9. Stop

### PSEUDOCODE

```
DECLARE FLOAT x1,y1,x2,y2,x3,y3
DECLARE FLOAT slope_xy,slope_yz,slope_xz
INPUT x1,y1,x2,y2,x3,y3
ASSIGN (y2-y1)/(x2-x1) to slope_xy
ASSIGN (y3-y2)/(x3-x2) to slope_yz
ASSIGN (y3-y1)/(x3-x1) to slope_xz
IF slope_xy = slope_yz AND slope_yz = slope_xz
    DISPLAY "Collinear points"
ELSE
    DISPLAY "Non-collinear points"
ENDIF
```



## FLOWCHART



### PROBLEM 3.7

Given the coordinates  $C(x,y)$  of the center of the circle and its radius  $r$ , write a program that will determine whether a point  $P(h,k)$  lies inside the circle, on the circle, or outside the circle.

**Distance between  $P(x,y)$  and centre  $C(h,k)$  is:**

$$D = \sqrt{(x - h)^2 + (y - k)^2}$$

- > If  $D < r$ , the point lies inside the circle.
- > If  $D = r$ , the point lies on the circle.
- > If  $D > r$ , the point lies outside the circle.

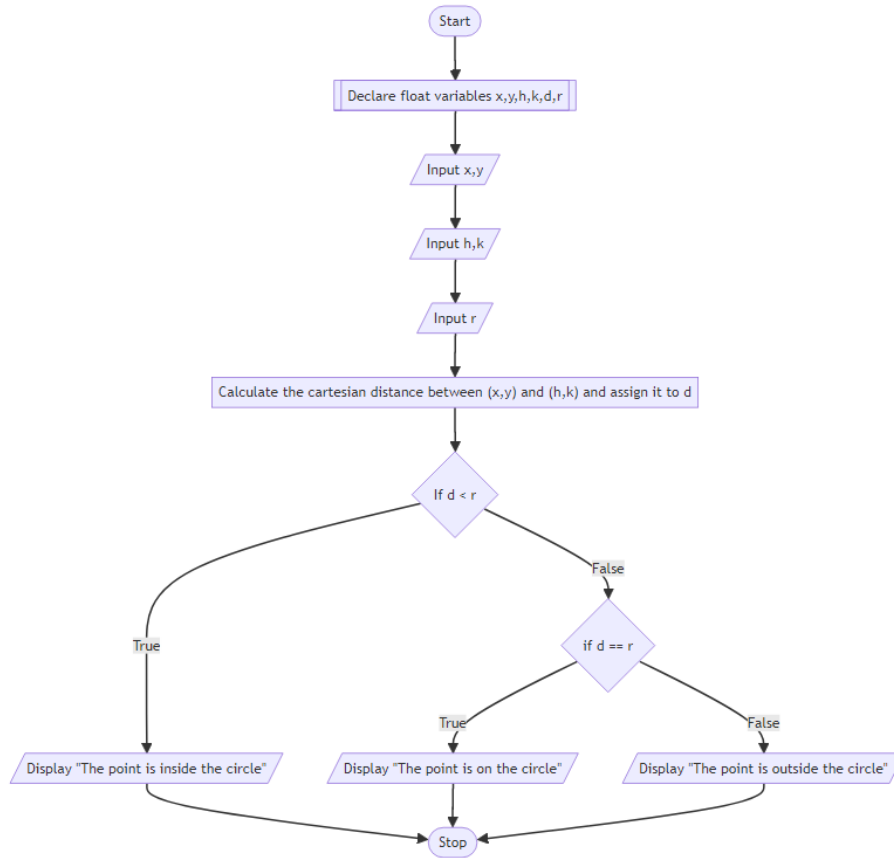
### ALGORITHM

1. Start
2. Declare float variables  $x,y,h,k,d,r$
3. Input  $x,y$
4. Input  $h,k$
5. Input  $r$
6. Calculate the cartesian distance between  $(x,y)$  and  $(h,k)$  and assign it to  $d$
7. If  $d < r$ , display "The point is inside the circle"
8. Else If  $d = r$ , display "The point is on the circle"
9. Else, display "The point is outside the circle"
10. Stop

### PSEUDOCODE

```
DECLARE FLOAT x,y,h,k,d,r
INPUT x,y
INPUT h,k
INPUT r
ASSIGN sqrt((x-h)^2 + (y-k)^2) to d
IF d < r
    DISPLAY "The point is inside the circle"
ELSE IF d = r
    DISPLAY "The point is on the circle"
ELSE
    DISPLAY "The point is outside the circle"
ENDIF
```

## FLOWCHART



## PROBLEM 3.8

Given a point  $(x,y)$ , write a program to find out if it lies on X-axis, Y-axis or origin.

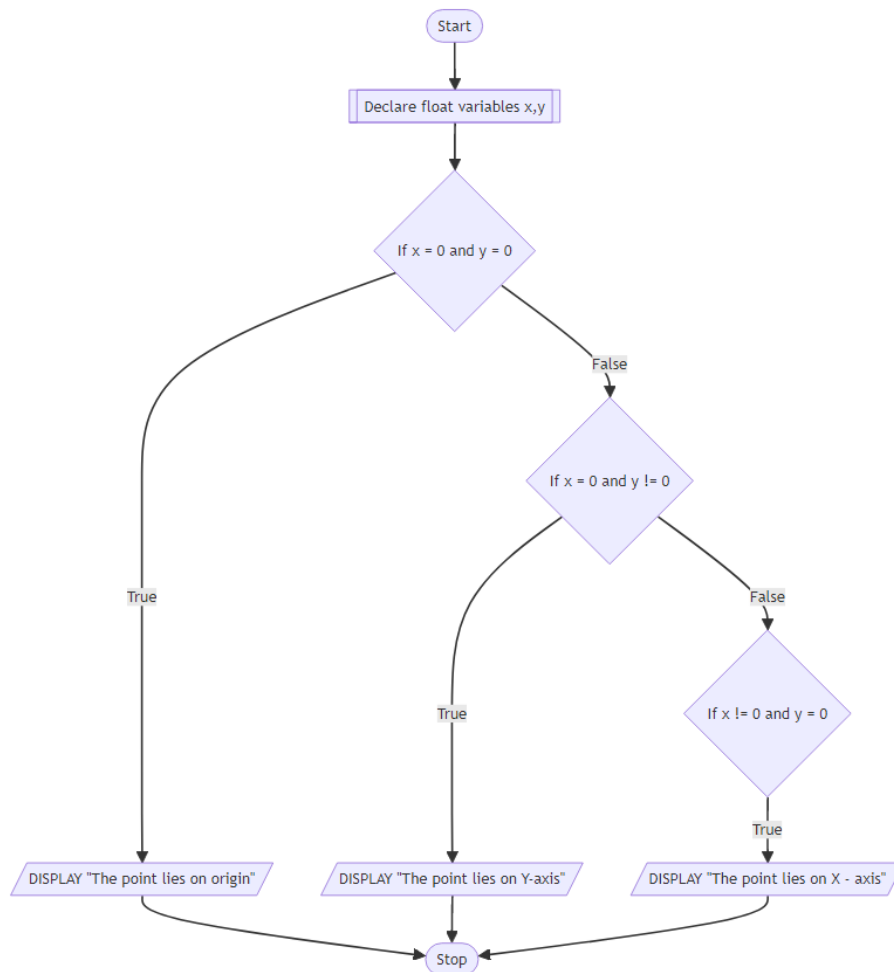
### ALGORITHM

1. Start
2. Declare float variables x and y
3. If  $x == 0$  and  $y == 0$ , display "The point lies on origin"
4. Else if  $x = 0$  and  $y != 0$ , display "The point lies on Y-axis"
5. Else if  $x != 0$  and  $y = 0$ , display "The point lies on X-axis"
6. Stop

## PSEUDOCODE

```
DECLARE FLOAT x,y
IF x = 0 AND y = 0
    DISPLAY "The point lies on origin"
ELSE IF x = 0 AND y != 0
    DISPLAY "The point lies on Y - axis"
ELSE IF x != 0 AND y = 0
    DISPLAY "The point lies on X - axis"
ENDIF
```

## FLOWCHART



### PROBLEM 3.9

According to Gregorian Calendar, it was Monday on the date 01/01/01. If any year is input through the keyboard, write a program to find out what is the day on 1st January of this year.

Year = 200

1st january 01 to 1st january 200 =  $365 \times 200$  (without considering leap years) + DAYS IN LEAP YEARS

odd days =  $365 \times 200 + \text{DAYS IN LEAP YEARS} / 7$

No.of odd days	Week day
0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

### ALGORITHM

1. Start
2. Declare the integer variables year, odd\_days
3. Input year
4. Create a helper function is\_leap(year) to determine whether the given year is a leap year or not
5. Run a loop from integer i = 0 to year-1
6. In that loop, check if each i is a leap year or not using the helper function defined above.
7. If the condition is true, add 366 to odd\_days
8. Else, add 365 to odd\_days
9. After the loop, assign the remainder of odd\_days and 7 back to odd\_days
10. Declare an string array weekdays with the days of week from monday to sunday
11. Display “The day of the week at 01.01.year is weekdays[odd\_days]”
12. Stop

### PSEUDOCODE

```
DECLARE INTEGER odd_days, year
INPUT year
ASSIGN 0 to odd_days
FUNCTION is_leap(year)
```

```

    IF REMAINDER(year,4) = 0 AND REMAINDER(year,100) != 0 OR REMAINDER(year,400) = 0
        RETURN True
    ELSE
        RETURN False
    ENDIF
ENDFUNCTION

FUNCTION main
    FOR INTEGER i = 1 to year-1
        IF is_leap(year) == True
            ADD 366 to odd_days
        ELSE
            ADD 365 to odd_days
        ENDIF
    ENDFOR
    ASSIGN odd_days%7 to odd_days

    DECLARE STRING ARRAY weekdays
    ASSIGN ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"] to a

    DISPLAY weekdays[odd_days]

ENDFUNCTION

```

## FLOWCHART

