# EXPERIMENT 3.1 : CONDITIONAL STATEMENTS

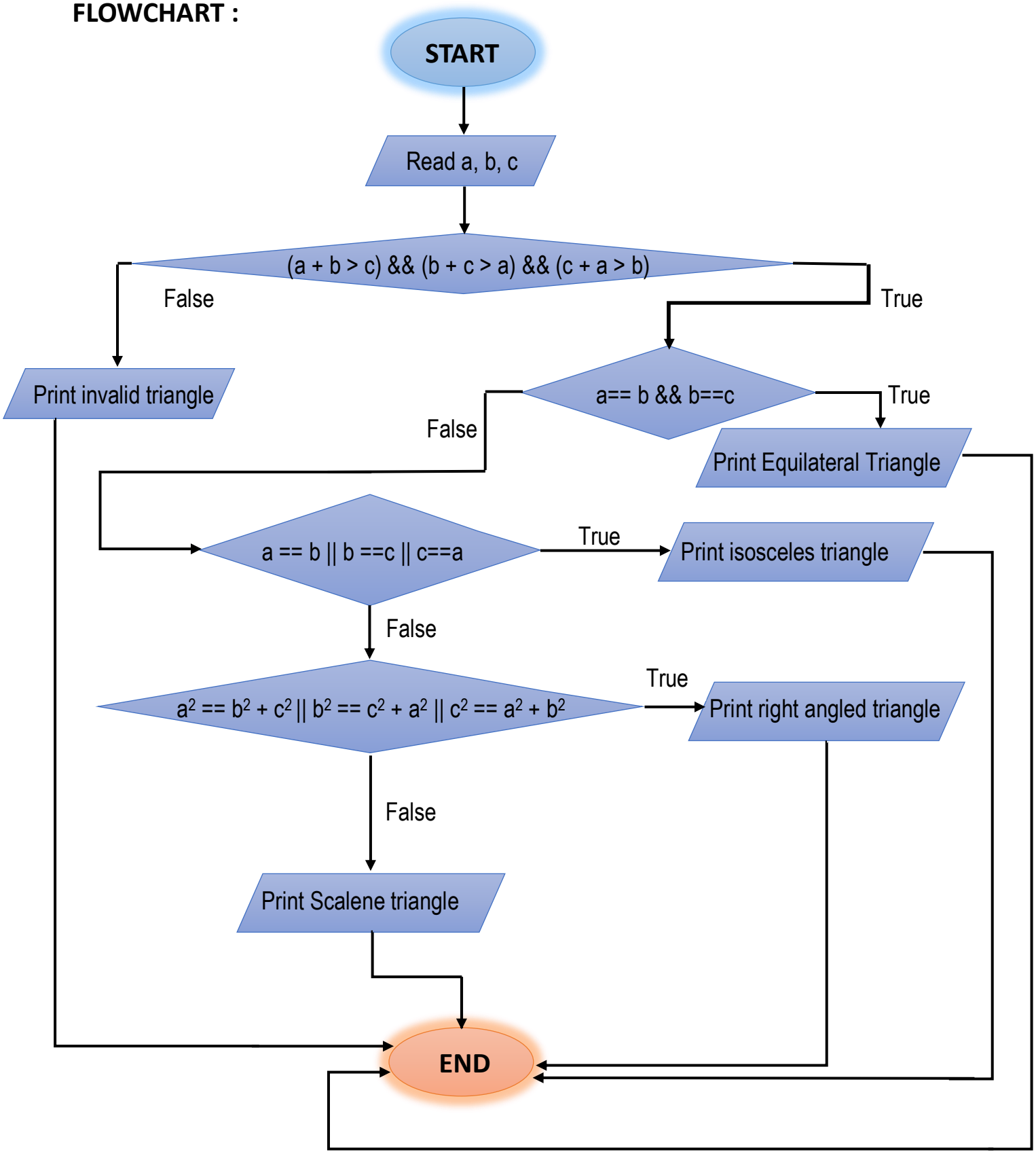
**Activity 1**: *WAP to take the check if the given triangle is valid or not. If the validity is established, do check if the triangle is isosceles , equilateral, right angle or scalene. Take sides of the triangle as input from the user.*

**Algorithm :**

STEP1: Start
STEP2: Read three sides a, b, c
STEP2: if (a + b > c) && (a + c > b) && (b + c > a) go to STEP 4
else
print "Triangle is not valid" and go to STEP 8
STEP4: If a==b && b==c then
print Equilateral triangle and go to Step 8
else
go to STEP 5
STEP5: Else if a==b || b==c || c==a then
print Isosceles triangle and go to STEP 8
else go to STEP 6
STEP6: Else if (a*a ==b*b + c*c) || (b*b == a*a + c*c) || (c*c == a*a + b*b)
print Right angled triangle and go to STEP 8
else go to STEP 7

STEP7: Else print "Scalene triangle"
STEP8: End

**FLOWCHART :**


START
Read a, b, c
(a + b > c) && (b + c > a) && (c + a > b)
False
True
Print invalid triangle
a== b && b==c
True
False
Print Equilateral Triangle
a == b || b ==c || c==a
True
Print isosceles triangle
False
a² == b² + c² || b² == c² + a² || c² == a² + b²
True
Print right angled triangle
False
Print Scalene triangle
END

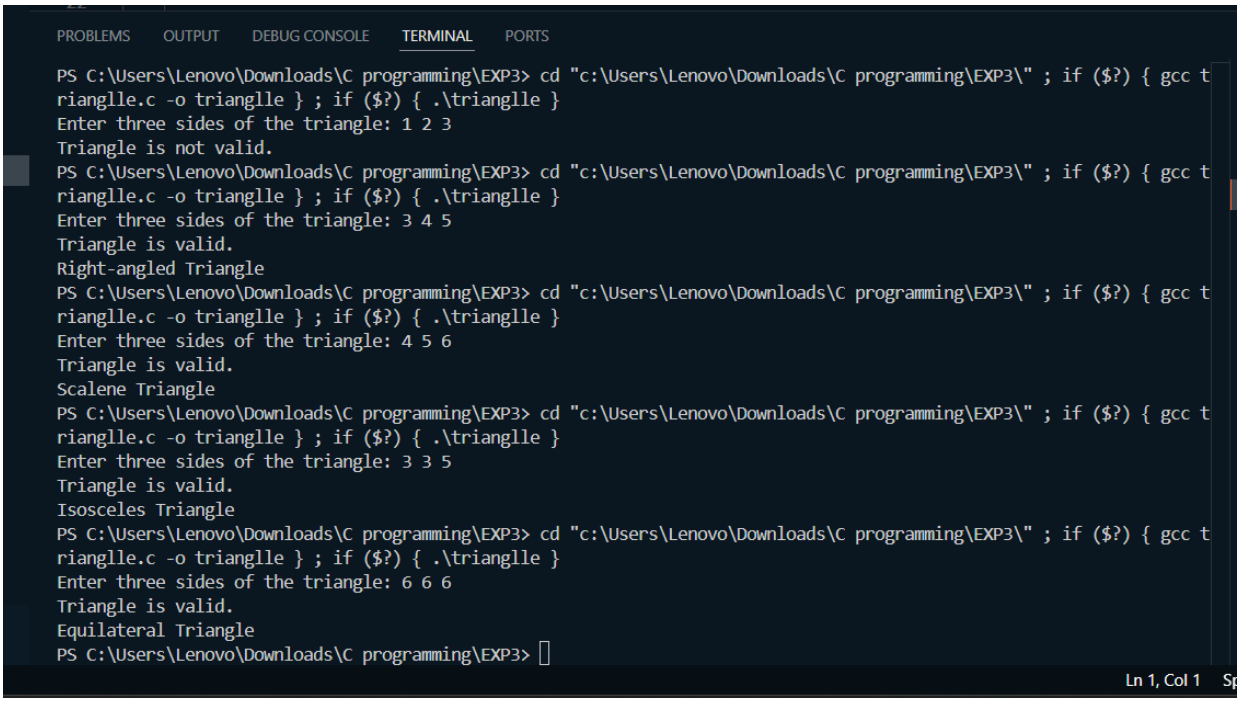
## PSEUDOCODE:

```
START

declare a, b, c AS float

print "Enter three sides of the triangle:"
input a, b, c

IF (a + b > c) AND (a + c > b) AND (b + c > a) THEN
    print "Triangle is valid."

    IF (a == b) AND (b == c) THEN
        print "Equilateral Triangle"
    ELSE IF (a == b) OR (b == c) OR (a == c) THEN
        print "Isosceles Triangle"
    ELSE IF (a*a == b*b + c*c) OR (b*b == a*a + c*c) OR (c*c == a*a + b*b) THEN
        print "Right-angled Triangle"
    ELSE
        print "Scalene Triangle"
    END IF

ELSE
    print "Triangle is not valid."
END IF

END
```

**CODE :**

```c
#include <stdio.h>

int main() {
    float a, b, c;


    printf("Enter three sides of the triangle: ");
    scanf("%f %f %f", &a, &b, &c);


    if ((a + b > c) && (a + c > b) && (b + c > a)) {
        printf("Triangle is valid.\n");


        if (a == b && b == c) {
            printf("Equilateral Triangle\n");
        }

        else if (a == b || b == c || a == c) {
            printf("Isosceles Triangle\n");
        }

        else if ((a*a == b*b + c*c) || (b*b == a*a + c*c) || (c*c == a*a + b*b)){
            printf("Right-angled Triangle\n");
        }

        else {
            printf("Scalene Triangle\n");
        }
    }
    else {
        printf("Triangle is not valid.\n");
    }

    return 0;
}
```

**OUTPUT:**




---

**Activity 2**: *WAP to compute the BMI index of the person and print the BMI values as per the following ranges. You can use the following formula to compute* $BMI = \frac{weight\ (kgs)}{height(m) * height(m)}$

| Body State | BMI |
|---|---|
| Starvation | < 15 |
| Anorexic | 15.1 to 17.5 |
| Underweight | 17.6 to 18.5 |
| Ideal | 18.6 to 24.9 |
| Overweight | 25 to 25.9 |
| Obese | 30 to 39.9 |

**ALGORITHM :**

**STEP1:** Start

**STEP2:** Declare variables weight, height, bmi

**STEP3:** Read weight and height

**STEP4:** Calculate bmi = weight / (height * height)

**STEP5:** If bmi < 15 then print "Starvation" and go to STEP11
else go to STEP6

**STEP6:** Else if bmi >= 15.1 && bmi <= 17.5 print "Anorexic" and goto STEP11
else go to STEP7

**STEP7:** Else if bmi >= 17.6 && bmi <= 18.5 print "Underweight" and go to STEP11
else go to STEP8

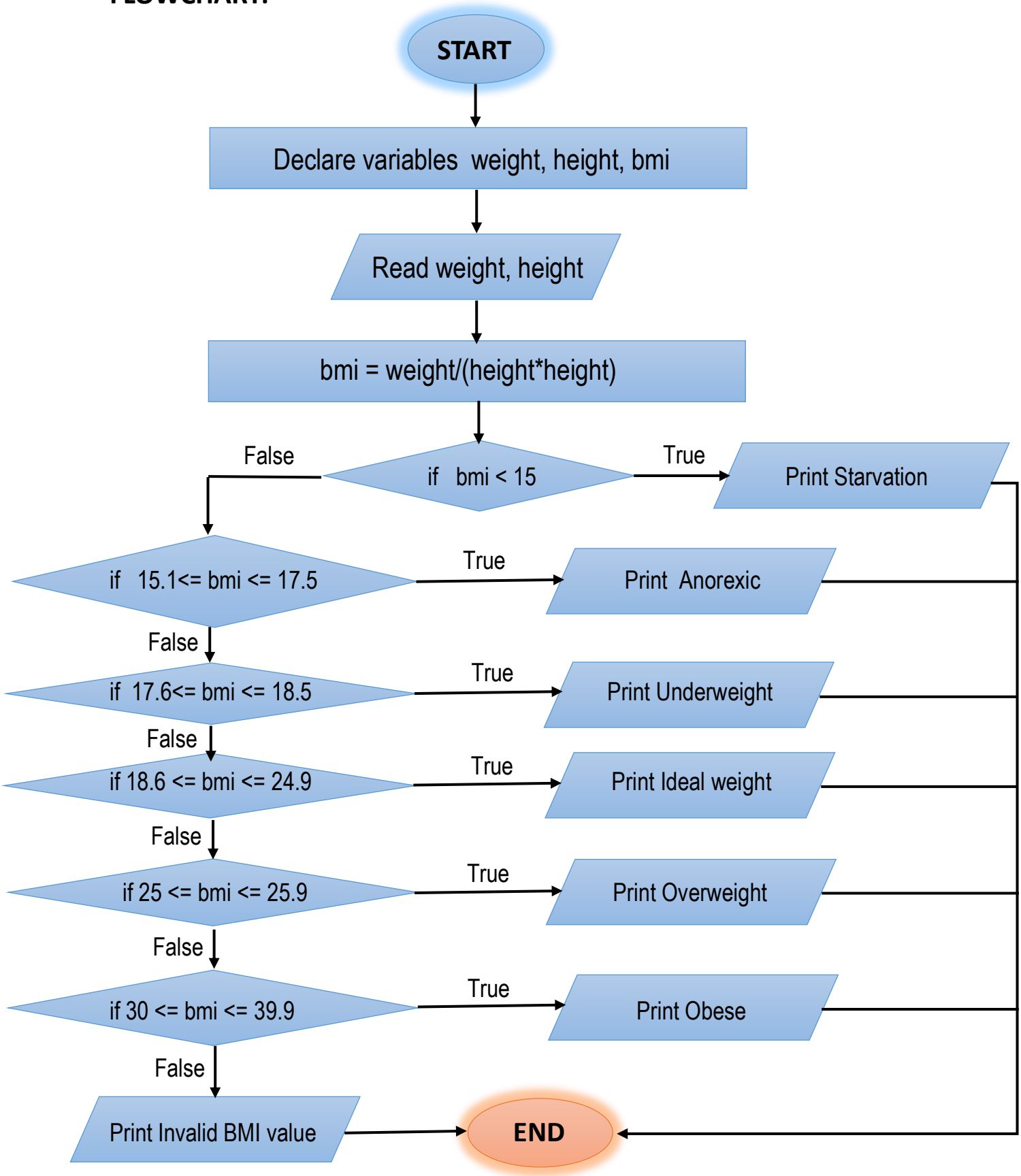
**STEP8:** Else if bmi >= 18.6 && bmi <= 24.9 print "Ideal" and go to STEP11
else go to STEP9

**STEP9:** Else if bmi >= 25 && bmi <= 25.9 print "Overweight" and go to STEP11
else go to STEP10

**STEP10:** Else if bmi >= 30 && bmi <= 39.9 "Obese" and go to STEP11
else print Invalid BMI value

**STEP11:** End

**FLOWCHART:**


START
Declare variables weight, height, bmi
Read weight, height
bmi = weight/(height*height)
if bmi < 15
False
True
Print Starvation
if 15.1<= bmi <= 17.5
True
Print Anorexic
False
if 17.6<= bmi <= 18.5
True
Print Underweight
False
if 18.6 <= bmi <= 24.9
True
Print Ideal weight
False
if 25 <= bmi <= 25.9
True
Print Overweight
False
if 30 <= bmi <= 39.9
True
Print Obese
False
Print Invalid BMI value
END

## PSEUDOCODE:

```
START

declare weight, height, bmi AS float

print "Enter weight (in kg):"
input weight

print "Enter height (in meters):"
input height

SET bmi = weight / (height * height)

IF bmi < 15 THEN
    print "Starvation"
ELSE IF bmi >= 15.1 AND bmi <= 17.5 THEN
    print "Anorexic"
ELSE IF bmi >= 17.6 AND bmi <= 18.5 THEN
    print "Underweight"
ELSE IF bmi >= 18.6 AND bmi <= 24.9 THEN
    print "Ideal weight"
ELSE IF bmi >= 25 AND bmi <= 29.9 THEN
    print "Overweight"
ELSE IF bmi >= 30 AND bmi <= 39.9 THEN
    print "Obese"
ELSE
    print "Invalid BMI value"
END IF

END
```

**CODE :**

```c
#include <stdio.h>

int main() {

    float weight, height, bmi;


    printf("Enter weight (in kg): ");
    scanf("%f", &weight);
    printf("Enter height (in meters): ");
    scanf("%f", &height);


    bmi = weight / (height * height);


    if (bmi < 15) {
        printf("Starvation\n");
    } else if (bmi >= 15.1 && bmi <= 17.5) {
        printf("Anorexic\n");
    } else if (bmi >= 17.6 && bmi <= 18.5) {
        printf("Underweight\n");
    } else if (bmi >= 18.6 && bmi <= 24.9) {
        printf("Ideal weight\n");
    } else if (bmi >= 25 && bmi <= 29.9) {
        printf("Overweight\n");
    } else if (bmi >= 30 && bmi <= 39.9) {
        printf("Obese\n");
    } else {
        printf("Invalid BMI value\n");
    }

    return 0;
}
```

## OUTPUT :


---

**Activity 3**: *WAP to check if three ($x_1$, $y_1$), ($x_2$, $y_2$), ($x_3$, $y_3$) points are collinear or not.*

## Algorithm :

**STEP 1**: Start

**STEP 2:** Declare variables x1, y1, x2, y2, x3, y3,area

**STEP 3:** Read (x1, y1), (x2, y2), (x3, y3)

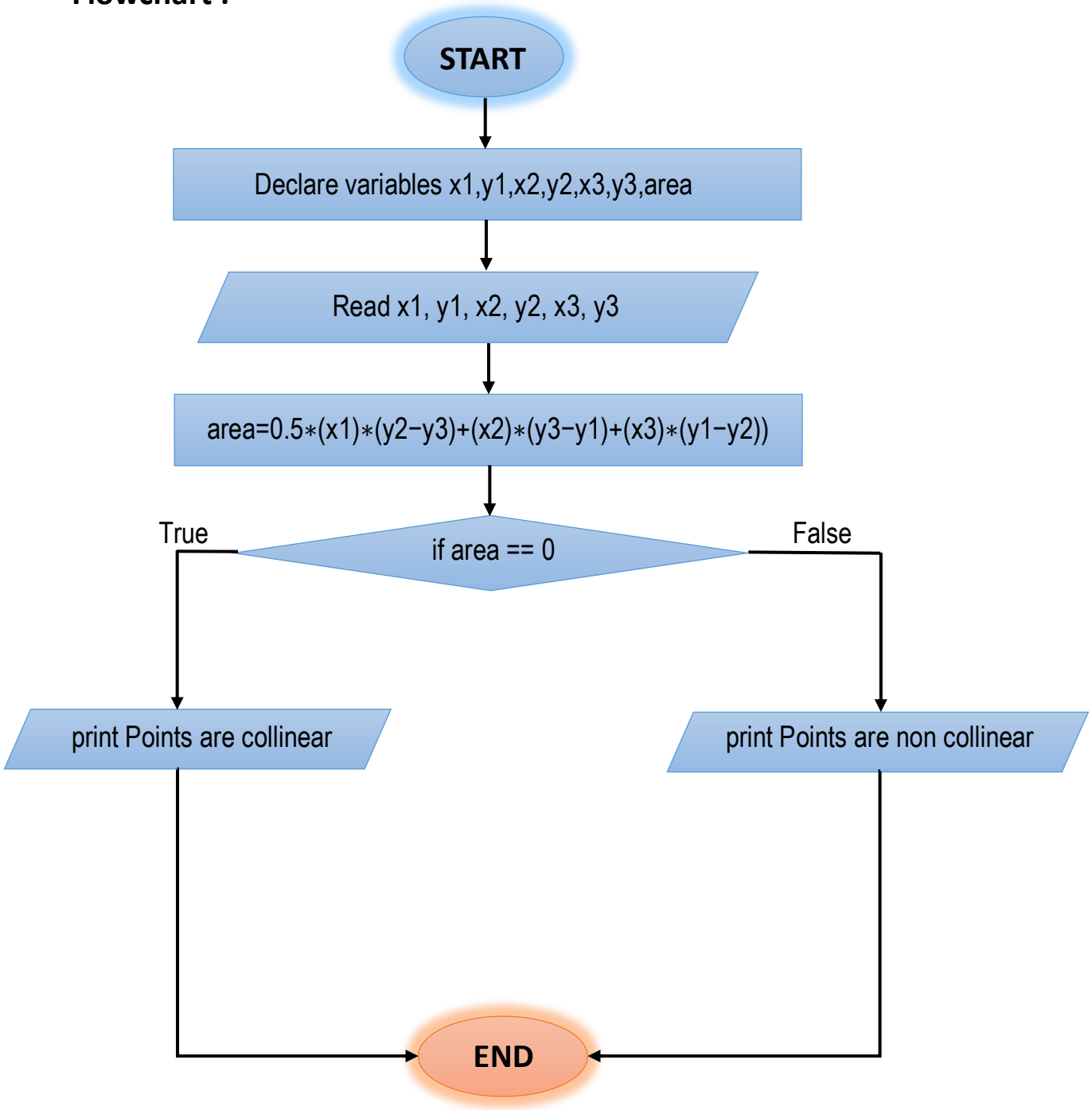
**STEP 4:** Calculate the area of triangle formed by the points
area=0.5∗(x1)∗(y2−y3)+(x2)∗(y3−y1)+(x3)∗(y1−y2))

**STEP 5:** if Area == 0 then print "Points are Collinear" and go to STEP7

**STEP 6:** else Print "Points are Not Collinear"

**STEP 7:** End

**Flowchart :**


START
Declare variables x1,y1,x2,y2,x3,y3,area
Read x1, y1, x2, y2, x3, y3
area=0.5∗(x1)∗(y2−y3)+(x2)∗(y3−y1)+(x3)∗(y1−y2))
True
if area == 0
False
print Points are collinear
print Points are non collinear
END

## PSEUDOCODE:

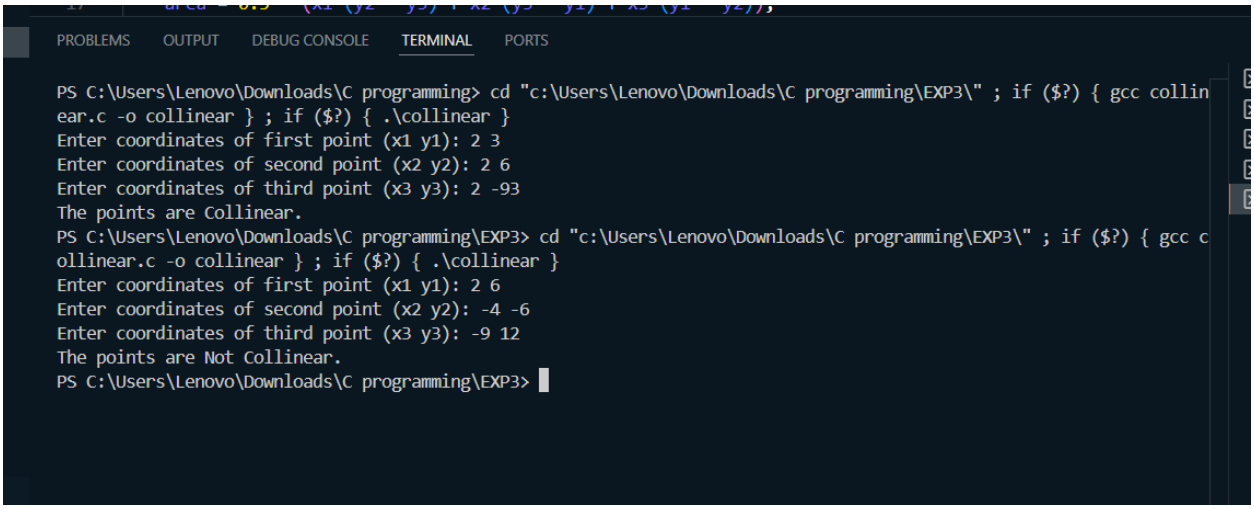
```
START

DECLARE x1, y1, x2, y2, x3, y3 AS integer
DECLARE area AS float

print "Enter coordinates of first point (x1 y1):"
input x1, y1

print "Enter coordinates of second point (x2 y2):"
input x2, y2

print "Enter coordinates of third point (x3 y3):"
input x3, y3

SET area = 0.5 * (x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2))

IF area == 0 THEN
    print "The points are Collinear."
ELSE
    print "The points are Not Collinear."
END IF

END
```

## CODE :

```c
#include <stdio.h>

int main() {

    int x1, y1, x2, y2, x3, y3;
    float area;

    printf("Enter coordinates of first point (x1 y1): ");
    scanf("%d %d", &x1, &y1);
```

```c
    printf("Enter coordinates of second point (x2 y2): ");
    scanf("%d %d", &x2, &y2);
    printf("Enter coordinates of third point (x3 y3): ");
    scanf("%d %d", &x3, &y3);

    area = 0.5 * (x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2));
    if (area == 0){
        printf("The points are Collinear.\n");
    }
    else{
        printf("The points are Not Collinear.\n");
    }
    return 0;
}
```

## OUTPUT :



**Activity 4:** *According to the 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.*

## ALGORITHM:

**STEP 1:** Start

**STEP 2:** Read year

**STEP 3:** Initialize total_days = 0

**STEP 4:** For i from 1 to year - 1 :
if (i % 4 == 0 && i % 100 != 0) || (i % 400 == 0) Leap year do total_days += 366
else Normal year do total_days += 365

**STEP 5:** Calculate day = total_days % 7

**STEP 6:** if day == 0 then print "Monday" and go to STEP 14
else go to STEP 7

**STEP 7:** elseif day == 1 then print "Tuesday" and go to STEP 14
else go to STEP 8

**STEP 8:** elseif day == 2 then print "Wednesday" and go to STEP 14
else go to STEP 9

**STEP 9:** elseif day == 3 then print "Thursday" and go to STEP 14
else go to STEP 10

**STEP 10:** elseif day == 4 then print "Friday" and go to STEP 14
else go to STEP 11

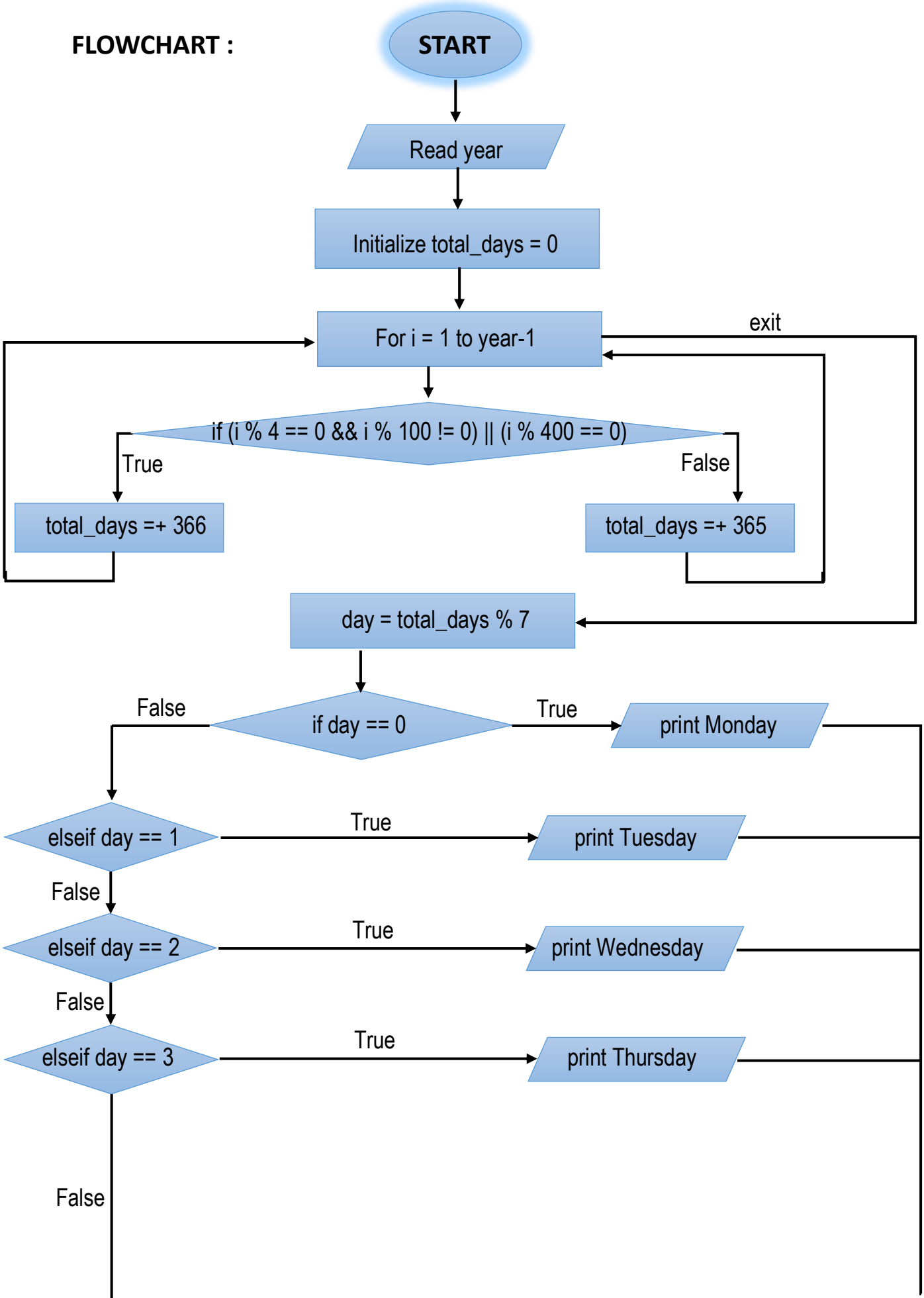
**STEP 11:** elseif day == 5 then print "Saturday" and go to STEP 14
else go to STEP 12

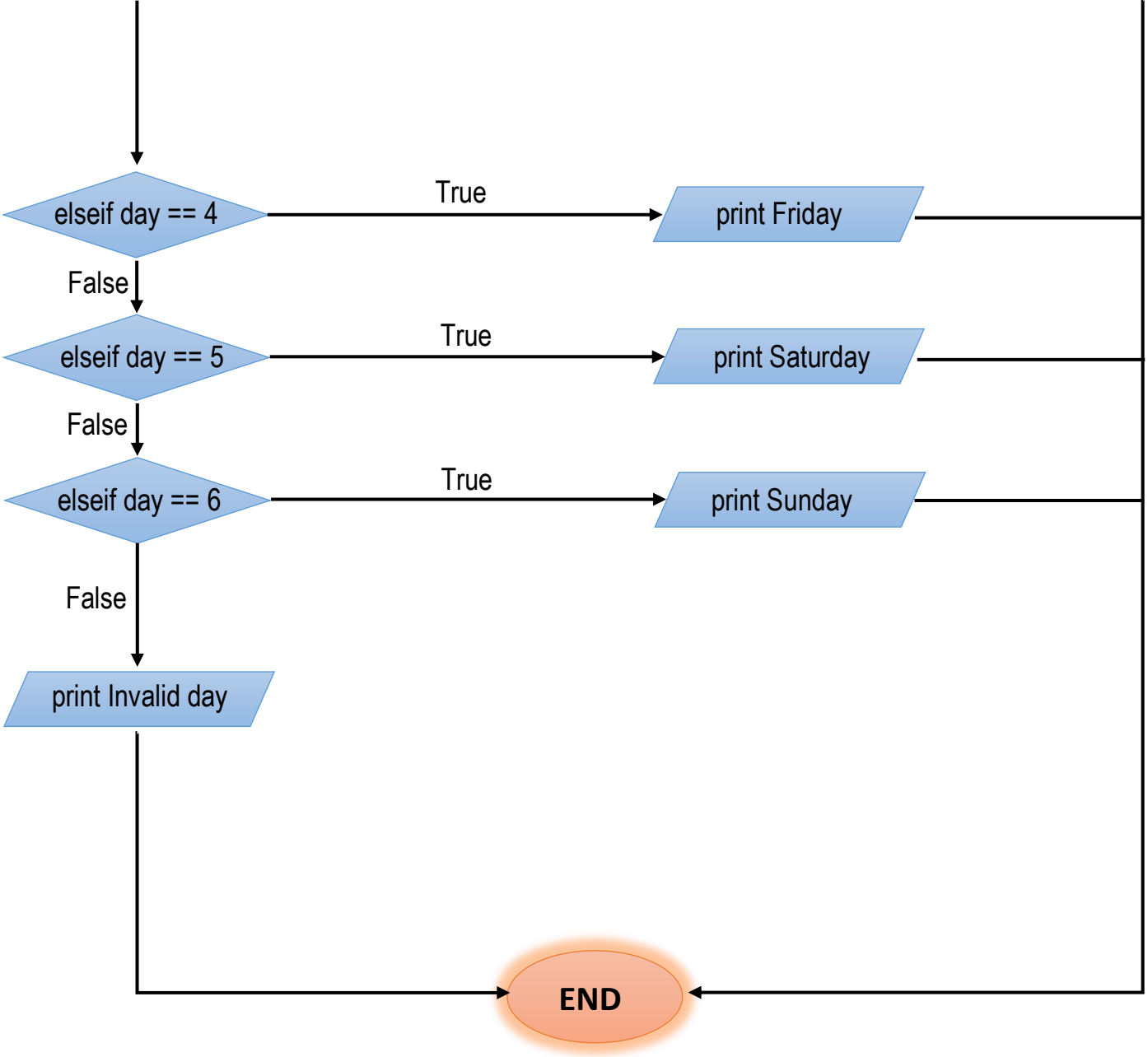
**STEP 12:** elseif day == 6 then print "Sunday" and go to STEP 14
else go to STEP 13

**STEP 13:** else print "Invalid day"

**STEP 14:** End

**FLOWCHART :**


START
Read year
Initialize total_days = 0
For i = 1 to year-1
exit
if (i % 4 == 0 && i % 100 != 0) || (i % 400 == 0)
True
False
total_days =+ 366
total_days =+ 365
day = total_days % 7
False
if day == 0
True
print Monday
elseif day == 1
True
print Tuesday
False
elseif day == 2
True
print Wednesday
False
elseif day == 3
True
print Thursday
False




**PSEUDOCODE:**

```
START

declare year, i, total_days, day AS integer
SET total_days = 0

print "Enter the year:"
input year

FOR i = 1 TO year - 1 DO
    IF ((i % 4 == 0 AND i % 100 != 0) OR (i % 400 == 0)) THEN
```

```
        total_days = total_days + 366
    ELSE
        total_days = total_days + 365
    END IF
END FOR

SET day = total_days % 7

IF day == 0 THEN
    print "Monday"
ELSE IF day == 1 THEN
    print "Tuesday"
ELSE IF day == 2 THEN
    print "Wednesday"
ELSE IF day == 3 THEN
    print "Thursday"
ELSE IF day == 4 THEN
    print "Friday"
ELSE IF day == 5 THEN
    print "Saturday"
ELSE IF day == 6 THEN
    print "Sunday"
ELSE
    print "Error"
END IF

END
```

## CODE :

```c
#include <stdio.h>

int main() {
    int year, i, total_days = 0, day;
    printf("Enter the year: ");
    scanf("%d", &year);

    for(i = 1; i < year; i++) {
        if((i % 4 == 0 && i % 100 != 0) || (i % 400 == 0)) {
            total_days += 366;  // Leap year
        } else {
            total_days += 365;  // Normal year
        }
```

```c
    }
    day = total_days % 7;
    if(day == 0){
        printf("Monday");
    }
    else if(day == 1){
        printf("Tuesday");
    }
    else if(day == 2){
        printf("Wednesday");
    }
    else if(day == 3){
        printf("Thursday");
    }
    else if(day == 4){
        printf("Friday");
    }
    else if(day == 5){
        printf("Saturday");
    }
    else if(day == 6){
        printf("Sunday");
    }
    else {
        printf("Error");
    }
    return 0;
}
```

## OUTPUT:

**Activity 5:** *WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles be three.*

**ALGORITHM:**

**STEP 1:** Start

**STEP 2:** Read l1, b1, l2, b2, l3, b3

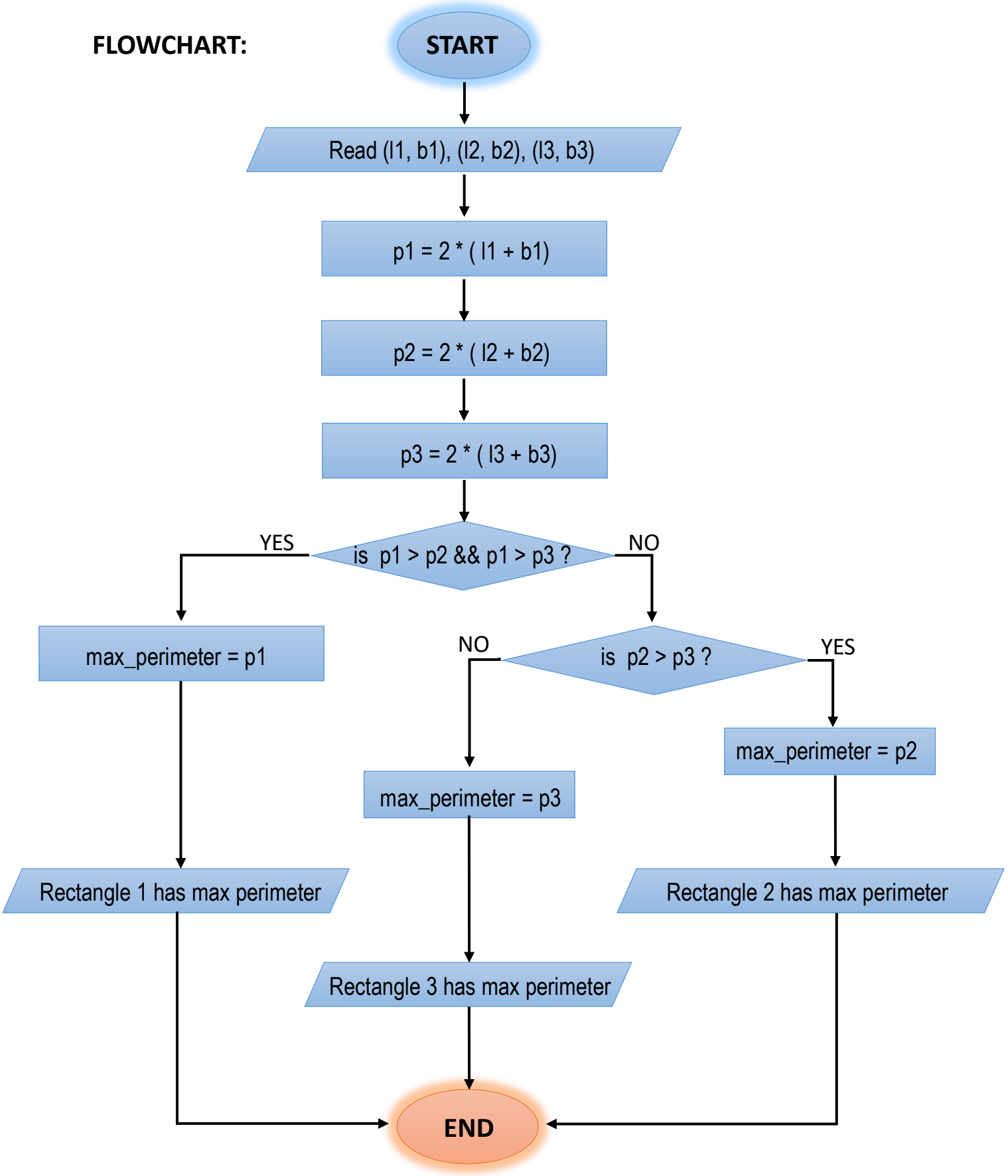
**STEP 3:** Calculate :

p1 = 2 * (l1 + b1)
p2 = 2 * (l2 + b2)
p3 = 2 * ( l3 + b3)

**STEP 4:** Using Ternary Operator :
**max_perimeter = (p1 > p2) ? ((p1 > p3) ? p1 : p3) : ((p2 > p3) ? p2 :p3)**
**STEP 5:** Display max_perimeter

**STEP 6 :** End

**FLOWCHART:**


START
Read (l1, b1), (l2, b2), (l3, b3)
p1 = 2 * ( l1 + b1)
p2 = 2 * ( l2 + b2)
p3 = 2 * ( l3 + b3)
is p1 > p2 && p1 > p3 ?
YES
NO
max_perimeter = p1
is p2 > p3 ?
NO
YES
max_perimeter = p3
max_perimeter = p2
Rectangle 1 has max perimeter
Rectangle 3 has max perimeter
Rectangle 2 has max perimeter
END

## PSEUDOCODE:

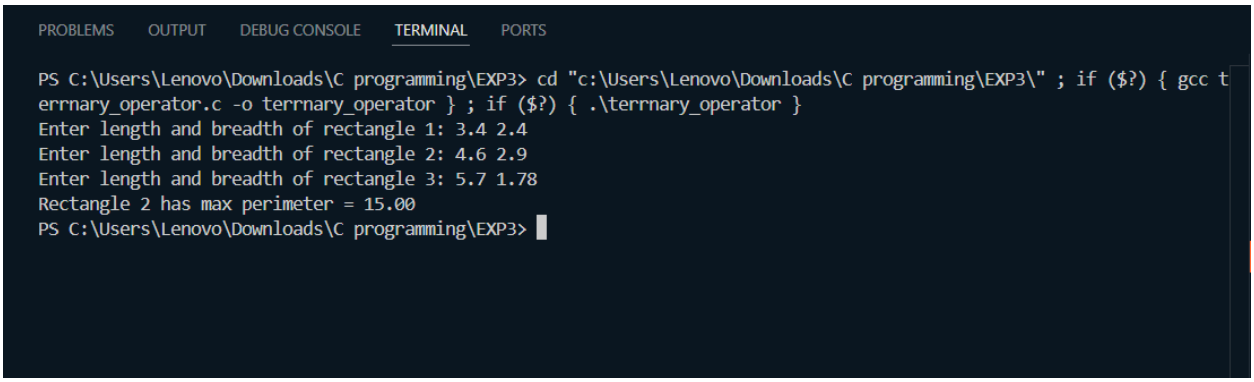
```
START

declare l1, b1, l2, b2, l3, b3 AS float
declare p1, p2, p3 AS float

print "Enter length and breadth of rectangle 1: "
input l1, b1

print "Enter length and breadth of rectangle 2: "
input l2, b2

print "Enter length and breadth of rectangle 3: "
input l3, b3

 p1 = 2 * (l1 + b1)
 p2 = 2 * (l2 + b2)
 p3 = 2 * (l3 + b3)

IF p1 > p2 AND p1 > p3 THEN
    print "Rectangle 1 has maximum perimeter = ", p1
ELSE IF p2 > p3 THEN
    print "Rectangle 2 has maximum perimeter = ", p2
ELSE
    print "Rectangle 3 has maximum perimeter = ", p3
END IF

END
```

## CODE:

```c
#include <stdio.h>

int main() {
    float l1, b1, l2, b2, l3, b3;
    float p1, p2, p3;

    printf("Enter length and breadth of rectangle 1: ");
    scanf("%f %f", &l1, &b1);
```

```c
    printf("Enter length and breadth of rectangle 2: ");
    scanf("%f %f", &l2, &b2);
    printf("Enter length and breadth of rectangle 3: ");
    scanf("%f %f", &l3, &b3);

    p1 = 2 * (l1 + b1);
    p2 = 2 * (l2 + b2);
    p3 = 2 * (l3 + b3);


    (p1 > p2 && p1 > p3) ? printf("Rectangle 1 has max perimeter = %.2f\n", p1) :
(p2 > p3 ?
         printf("Rectangle 2 has max perimeter = %.2f\n", p2) :
         printf("Rectangle 3 has max perimeter = %.2f\n", p3));

    return 0;
}
```

## OUTPUT:



# EXPERIMENT 3.2 : LOOPS

---

**ACTIVITY 1:** *WAP to enter numbers till the user wants. At the end it should display the count of positives, negatives and zeroes entered.*

**ALGORITHM :**

**STEP 1:** Start

**STEP 2:** Initialize pos = 0, neg = 0, zero = 0

**STEP 3:** Read num

**STEP 4:** if num > 0 then
pos = pos + 1 and go to **STEP 6**
else go to **STEP 5**

**STEP 5:** if num < 0 then
neg = neg + 1 and go to **STEP 6**
else
zero = zero + 1 and go to **STEP 6**

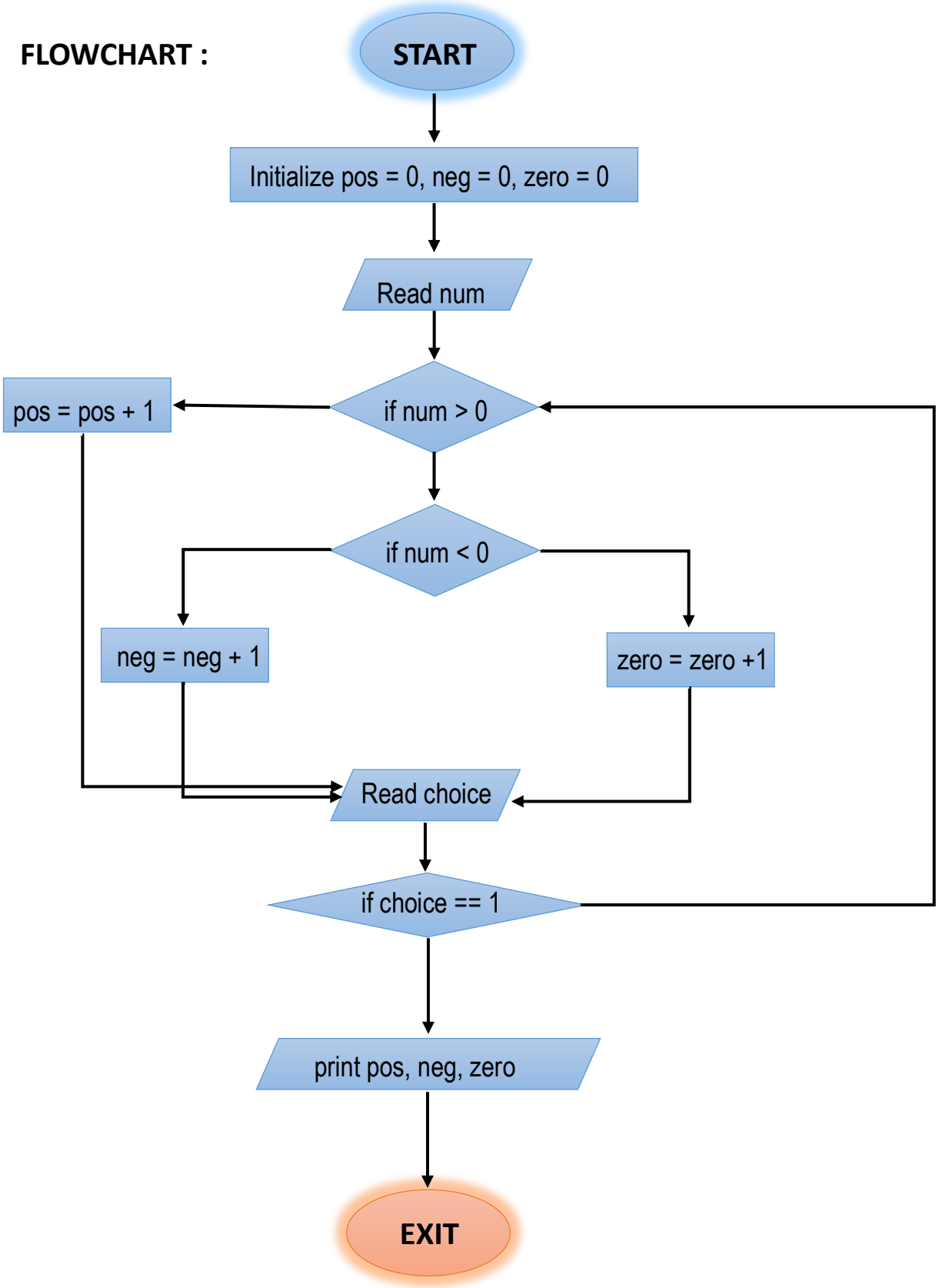
**STEP 6:** Read choice

**STEP 7:** if choice == 1 then go to **STEP 3**
else go to **STEP 8**

**STEP 8:** Print pos, neg, zero

**STEP 9:** End

**FLOWCHART :**


START
Initialize pos = 0, neg = 0, zero = 0
Read num
if num > 0
pos = pos + 1
if num < 0
neg = neg + 1
zero = zero +1
Read choice
if choice == 1
print pos, neg, zero
EXIT

## PSEUDOCODE :

```
START

 num, pos, neg, zero, choice as integer
SET pos = 0, neg = 0, zero = 0

REPEAT
    print "Enter a number:"
    input num

    IF num > 0 THEN
        pos = pos + 1
    ELSE IF num < 0 THEN
        neg = neg + 1
    ELSE
        zero = zero + 1
    END IF

    print "Do you want to continue? (1 = Yes, 0 = No):"
    input choice

UNTIL choice != 1

print "Count of positive numbers =", pos
print "Count of negative numbers =", neg
print "Count of zeroes =", zero

END
```

## CODE :

```
#include <stdio.h>

int main() {
    int num, pos = 0, neg = 0, zero = 0;
    int choice;

    do {
        printf("Enter a number: ");
```

```c
        scanf("%d", &num);

        if (num > 0)
            pos = pos + 1;
        else if (num < 0)
            neg = neg + 1;
        else
            zero = zero + 1;

        printf("Do you want to continue? (1 = Yes, 0 = No): ");
        scanf("%d", &choice);

    } while (choice == 1);

    printf("\nCount of positive numbers: %d", pos);
    printf("\nCount of negative numbers: %d", neg);
    printf("\nCount of zeroes: %d\n", zero);

    return 0;
}
```

## OUTPUT :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\Lenovo\Downloads\C programming> cd "c:\Users\Lenovo\Downloads\C programming\EXP3\" ; if ($?) { gcc count.
c -o count } ; if ($?) { .\count }
Enter a number: 4
Do you want to continue? (1 = Yes, 0 = No): 1
Enter a number: -67
Do you want to continue? (1 = Yes, 0 = No): 1
Enter a number: 0
Do you want to continue? (1 = Yes, 0 = No): 1
Enter a number: 684
Do you want to continue? (1 = Yes, 0 = No): 0

Count of positive numbers: 2
Count of negative numbers: 1
Count of zeroes: 1
PS C:\Users\Lenovo\Downloads\C programming\EXP3>
```

**ACTIVITY 2:** *WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting. (Num * 1 = Num)*

**ALGORITHM :**

**STEP 1:** Start

**STEP 2:** Read num

**STEP 3:** Initialize i = 1

**STEP 4:** Repeat **STEP 5** to **STEP 7** while i <= 10

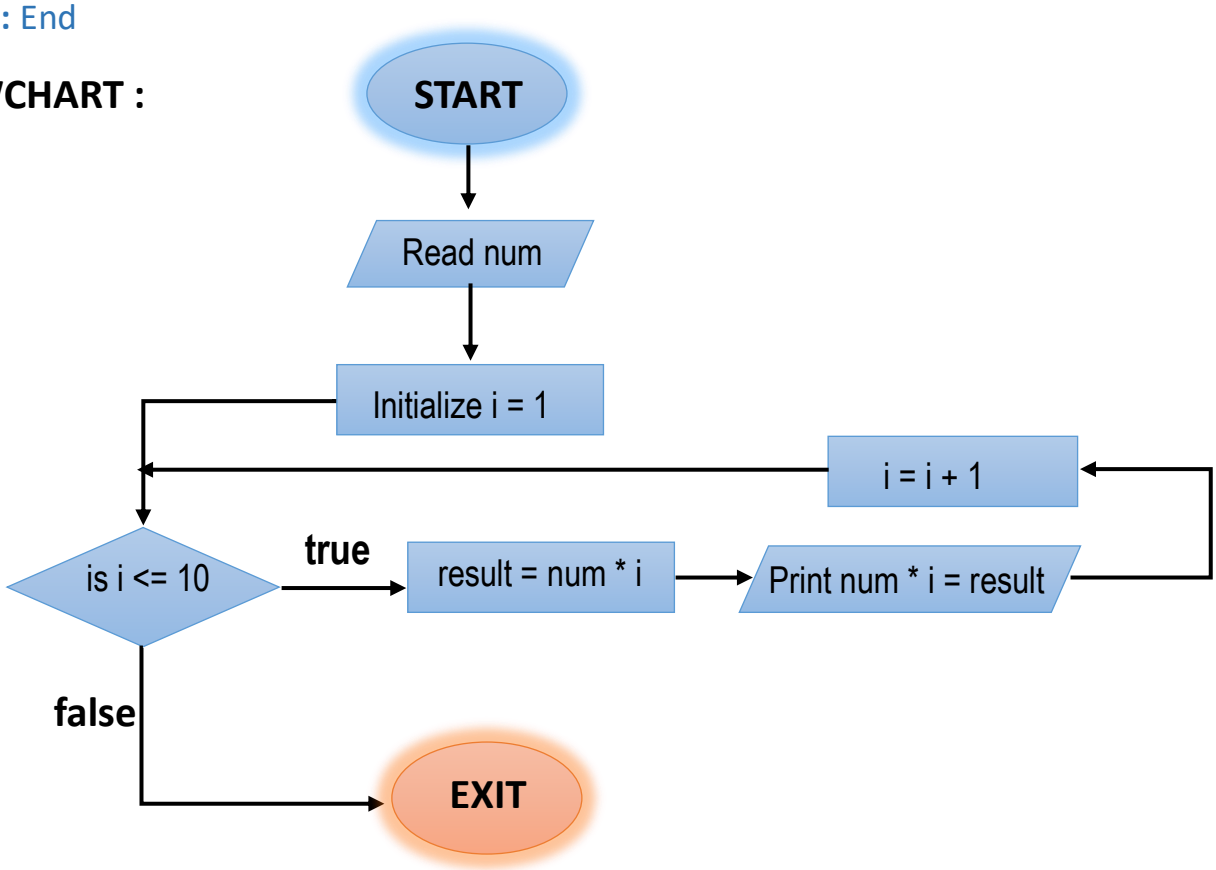
**STEP 5:** result = num * i

**STEP 6:** print num * i = result

**STEP 7:** i = i + 1

**STEP 8:** End

**FLOWCHART :**



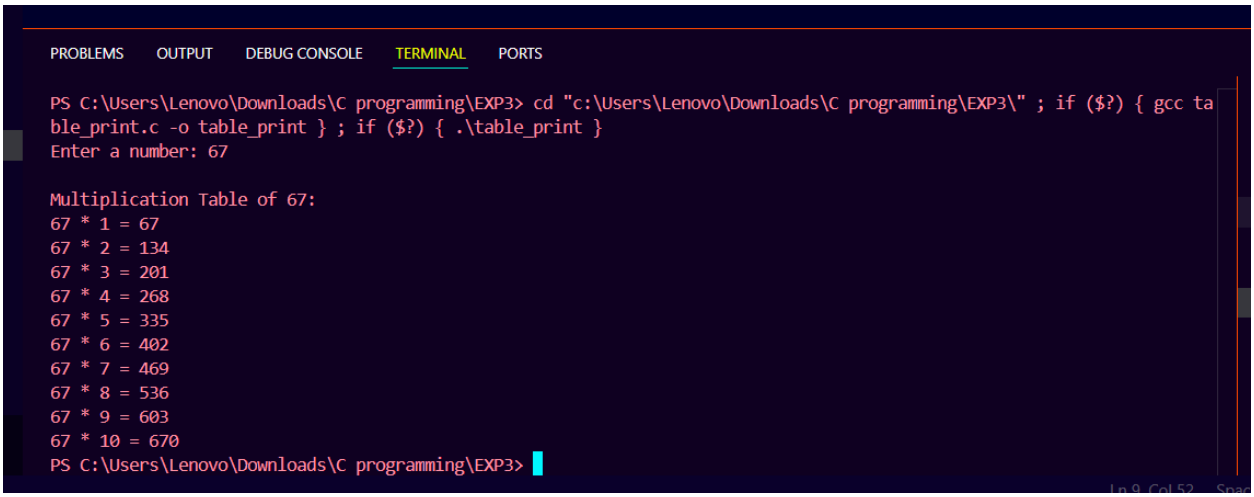
## PSEUDOCODE:

```
START

declare num, i, result as integer

print "Enter a number:"
input num

print "Multiplication Table of", num, ":"

FOR i = 1 TO 10 DO
    result = num * i
    print num, "*", i, "=", result
END FOR

END
```

## CODE :

```c
#include <stdio.h>

int main() {
    int num, i, result;

    printf("Enter a number: ");
    scanf("%d", &num);

    printf("\nMultiplication Table of %d:\n", num);

    for(i = 1; i <= 10; i++) {
        result = num * i;
        printf("%d * %d = %d\n", num, i, result);
    }

    return 0;
}
```

**OUTPUT :**




---

**ACTIVITY 3:** *WAP to generate the following set of output :*

*a.*

1

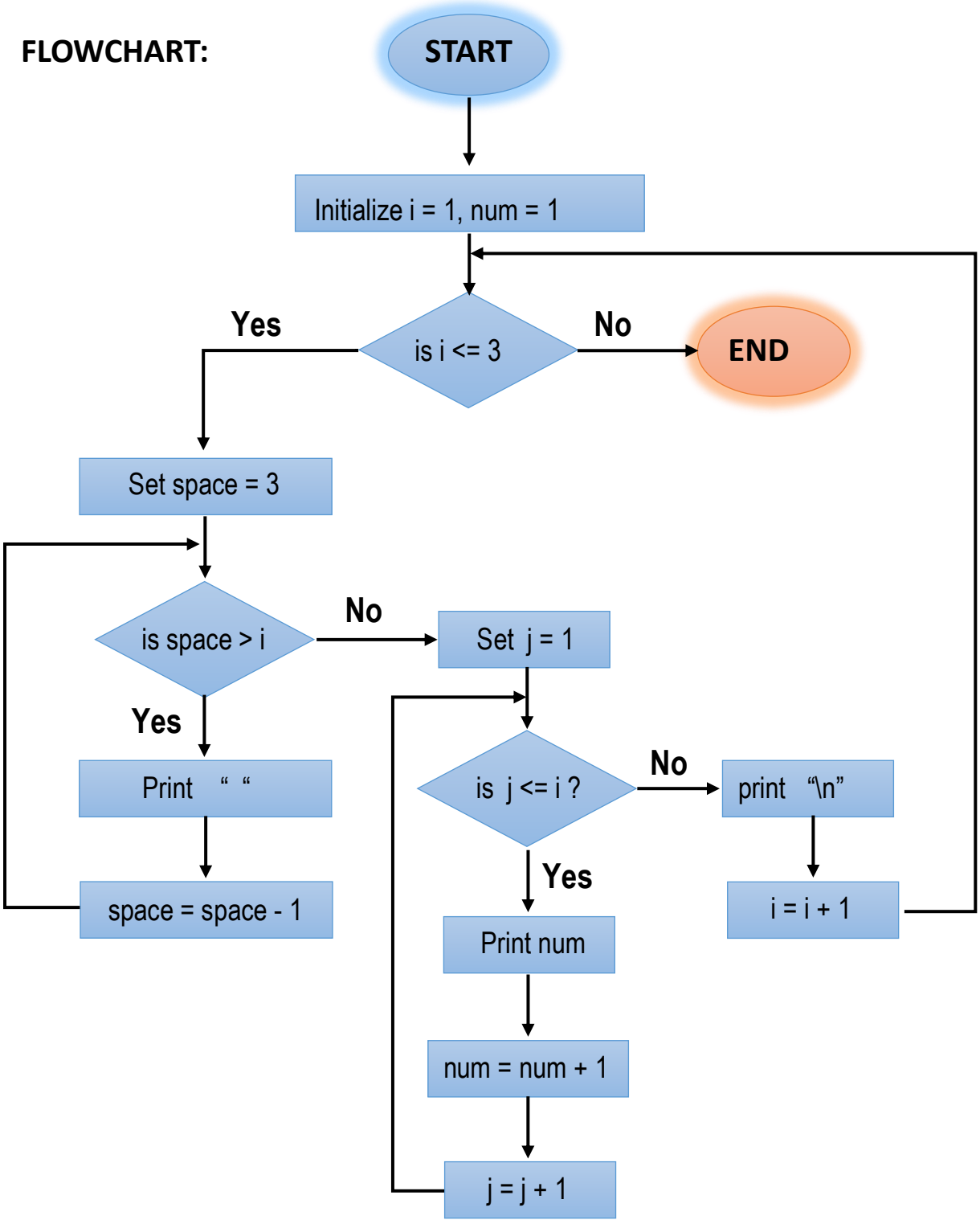
2 3

4 5 6

**ALGORITHM :**

**STEP 1:** Start
**STEP 2:** Initialize i = 1, num = 1
**STEP 3:** Repeat **STEP4** to **STEP9** while i <= 3
**STEP 4:** Set space = 3
**STEP 5:** Repeat while space > i
    Print " "
    space = space - 1
**STEP 6**: Set j = 1
**STEP 7:** Repeat while j <= i
    Print num
    num = num + 1
    j = j + 1
**STEP 8:** Print "\n"
**STEP 9:** i = i + 1
**STEP 10:** Stop

**FLOWCHART:**


START
Initialize i = 1, num = 1
is i <= 3
Yes
No
END
Set space = 3
is space > i
No
Yes
Set j = 1
Print " "
space = space - 1
is j <= i ?
No
Yes
print "\n"
i = i + 1
Print num
num = num + 1
j = j + 1

## PSEUDOCODE :

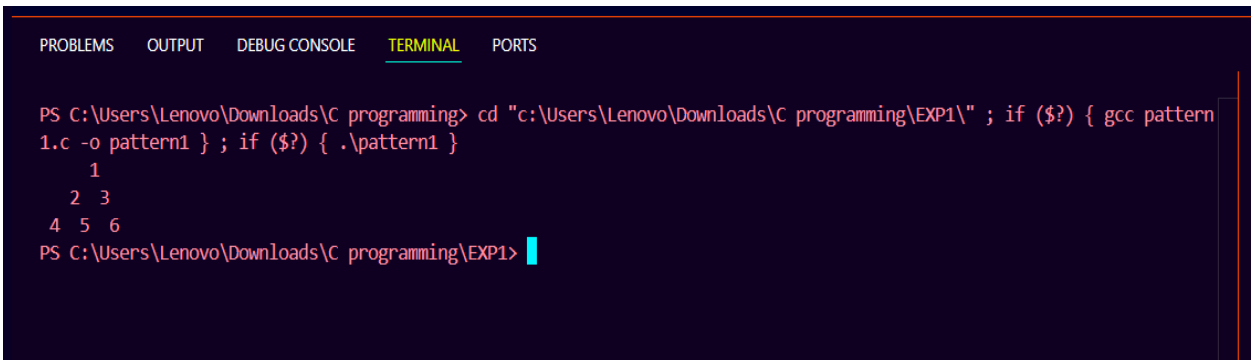
```
START

DECLARE i, j, space, num AS integer
SET num = 1

FOR i = 1 TO 3 DO
    FOR space = 3 TO i + 1 DO
        print "  "
    END FOR

    FOR j = 1 TO i DO
        print num
        num = num + 1
    END FOR

    print newline
END FOR

END
```

## CODE :

```c
#include <stdio.h>

int main() {
    int i, j, space, num = 1;

    for (i = 1; i <= 3; i++) {
        for (space = 3; space > i; space--) {
            printf("  ");
        }

        for (j = 1; j <= i; j++) {
            printf("%2d ", num);
            num++;
```

```
        }

        printf("\n");
    }

    return 0;
}
```

## OUTPUT:



b.

```
    1
   1 1
  1 2 1
 1 3 3 1
```

**ALGORITHM :**

**STEP 1:** Start

**STEP 2:** Initialize n = 4, i = 0

**STEP 3:** Repeat **STEP4** to **STEP15** while i < n else go to **STEP16**

**STEP 4:** Set space = 1

**STEP 5:** Repeat **STEP6** to **STEP7** while space <= n – i else go to **STEP8**

**STEP 6:** Print a space " "

**STEP 7:** space = space + 1

**STEP 8:** Set coef = 1

**STEP 9:** Set j = 0

**STEP 10:** Repeat **STEP11** to **STEP13** while j <= I else go to **STEP14**

**STEP 11:** Print coef

**STEP 12:** coef = coef * (i - j) / (j + 1)

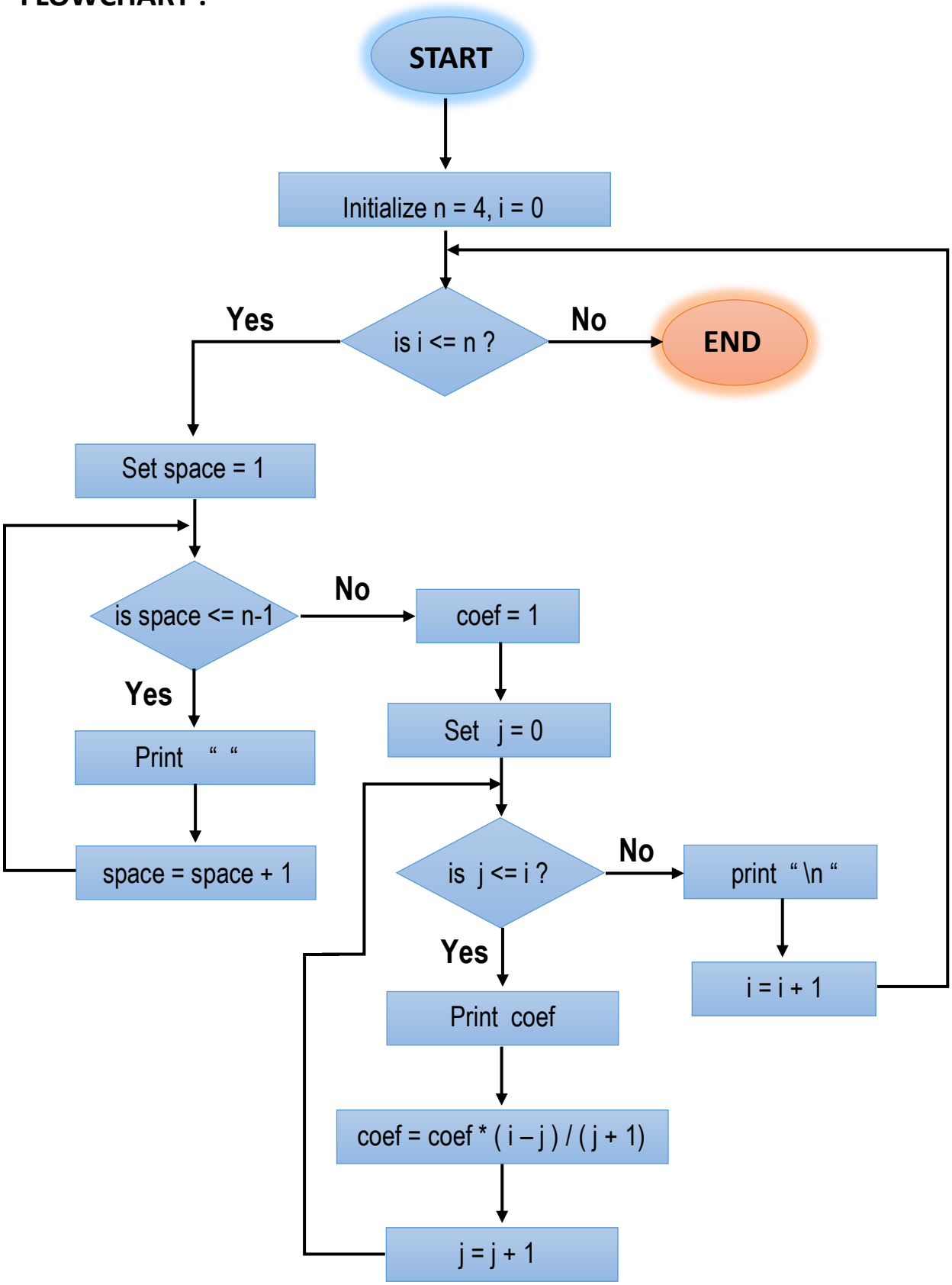
**STEP 13:** j = j + 1

**STEP 14:** Print " \n"

**STEP 15:** i = i + 1

**STEP 16:** Stop

**FLOWCHART :**


START
Initialize n = 4, i = 0
is i <= n ?
Yes
No
END
Set space = 1
is space <= n-1
No
Yes
coef = 1
Print " "
Set j = 0
space = space + 1
is j <= i ?
No
print " \n "
Yes
i = i + 1
Print coef
coef = coef * ( i – j ) / ( j + 1)
j = j + 1

## PSEUDOCODE :

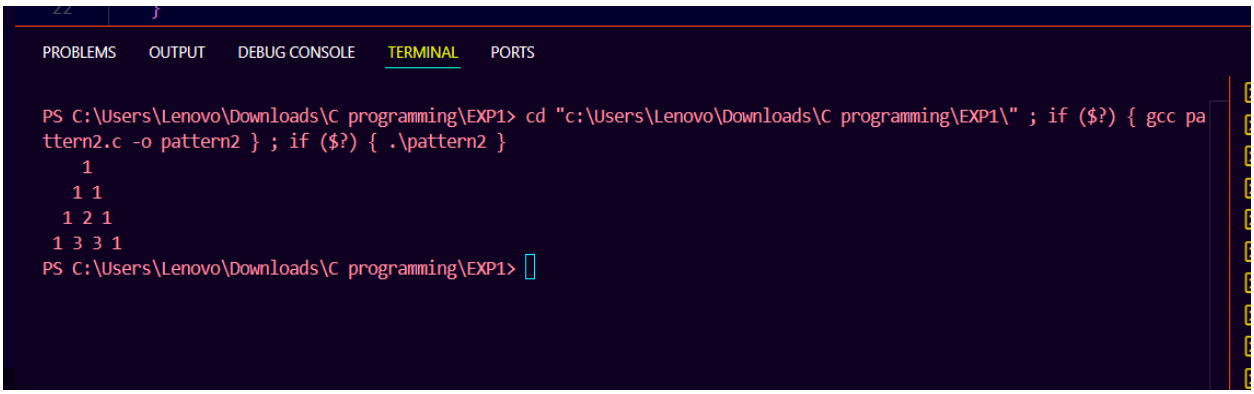
```
START
    SET n = 4
    SET i = 0

    WHILE i < n DO
        SET space = 1
        WHILE space <= n - i DO
            PRINT " "
            space = space + 1
        END WHILE

        SET coef = 1
        SET j = 0
        WHILE j <= i DO
            PRINT coef
            coef = coef * (i - j) / (j + 1)
            j = j + 1
        END WHILE

        PRINT new line
        i = i + 1
    END WHILE
END
```

## CODE :

```c
#include <stdio.h>

int main() {
    int n = 4;
    int i, j, space, coef;

    for (i = 0; i < n; i++) {

        for (space = 1; space <= n - i; space++) {
            printf(" ");
        }

        coef = 1;
        for (j = 0; j <= i; j++) {
            printf("%d ", coef);
            coef = coef * (i - j) / (j + 1);
```

```
        }

        printf("\n");
    }

    return 0;
}
```

**OUTPUT :**



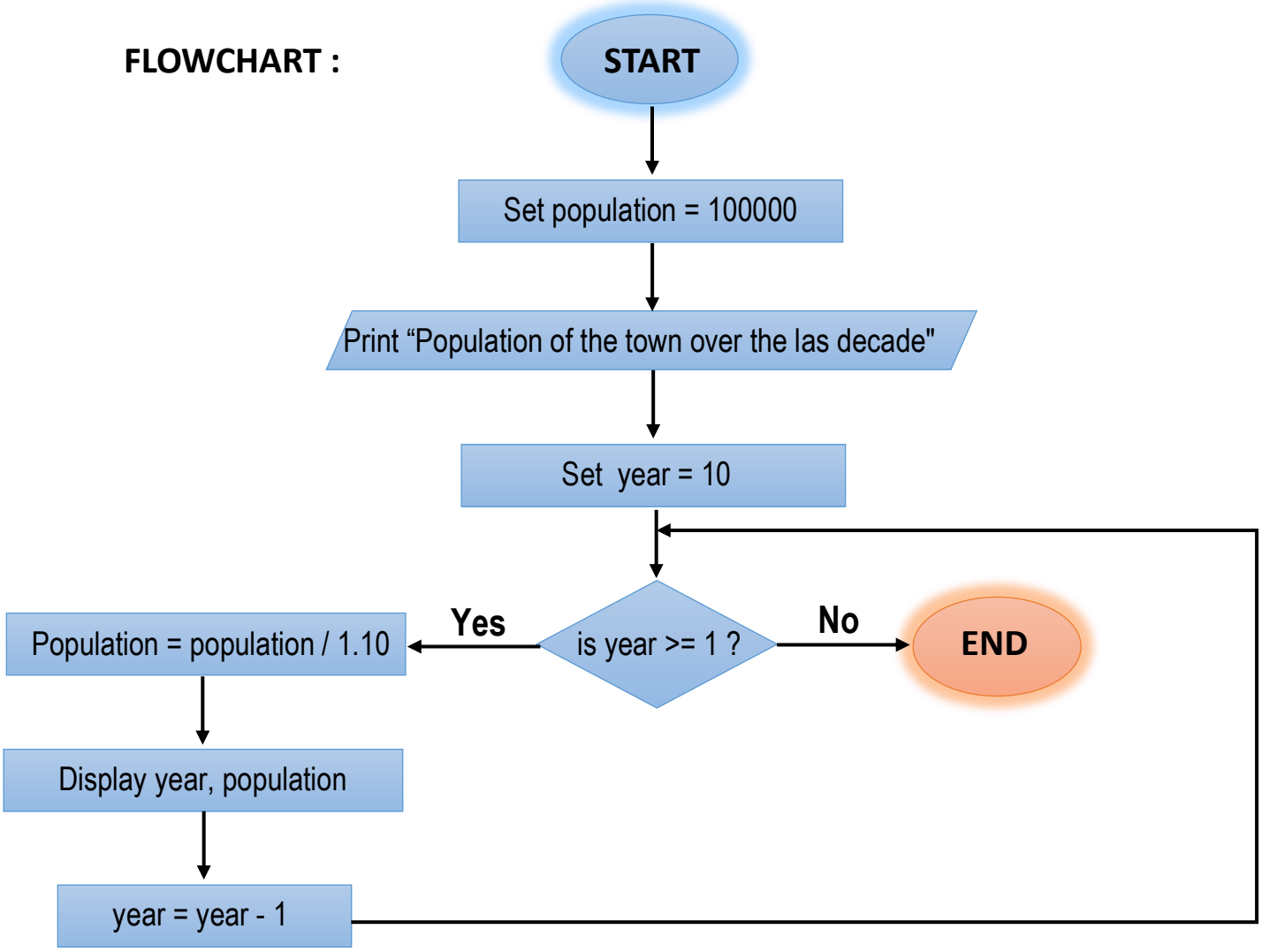

---

**Activity 4 :** *The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in the last decade.*

**ALGORITHM :**

**STEP 1:** Start
**STEP 2:** Initialize population = 100000
**STEP 3:** Display “Population of the town over the last 10 years:”
**STEP 4:** Set year = 10
**STEP 5:** if year >= 1 go to **STEP6** else go to **STEP9**
**STEP 6:** Compute population = population / 1.10
**STEP 7:** Display year, population
**STEP 8:** year = year - 1 and go to **STEP5**
**STEP 9:** Stop

**FLOWCHART :**



## PSEUDOCODE :

```
START

population = 100000
print "Population of the town over the last decade:"

FOR year = 10 TO 1 STEP -1 DO
    population = population / 1.10
    print "Year", year, "ago:", population
END FOR

END
```

## CODE :

```c
#include <stdio.h>

int main() {
    float population = 100000;

    printf("Population of the town over the last 10 years:\n");

    for (int year = 10; year >= 1; year--) {
        population = population / 1.10;  // reverse 10% growth
        printf("Year %d ago: %.0f\n", year, population);
    }

    return 0;
}
```

## OUTPUT :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Lenovo\Downloads\C programming> cd "c:\Users\Lenovo\Downloads\C programming\EXP1\" ; if ($?) { gcc populat
ion.c -o population } ; if ($?) { .\population }
Population of the town over the last 10 years:
Year 10 ago: 90909
Year 9 ago: 82645
Year 8 ago: 75131
Year 7 ago: 68301
Year 6 ago: 62092
Year 5 ago: 56447
Year 4 ago: 51316
Year 3 ago: 46651
Year 2 ago: 42410
Year 1 ago: 38554
PS C:\Users\Lenovo\Downloads\C programming\EXP1>
```

**Activity 5:** *Ramanujan Number is the smallest number that can be expressed as the sum of two cubes in two different ways. WAP to print all such numbers up to a reasonable limit.*

*for a number L = 20. (that is limit)*

*Example of Ramanujan number:* ***1729***

## ALGORITHM :

**STEP 1:**