Name Priyadarshi Prabhakar SAP ID 590029237

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

PSEUDOCODE:

```
START

Declare float a, b, c

Print "Enter three sides: "
Input a, b, c

// Check triangle validity
IF (a + b > c) AND (a + c > b) AND (b + c > a) THEN

    // Check Equilateral
    IF (a == b) AND (b == c) THEN
        Print "Equilateral triangle"

    ELSE
        // Check Isosceles
        IF (a == b) OR (b == c) OR (c == a) THEN
            Print "Isosceles triangle"

        ELSE
```

```
        // Check Right-angled
        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"
        ENDIF
      ENDIF
   ENDIF

ELSE
   Print "Triangle is not valid"
ENDIF

END
```

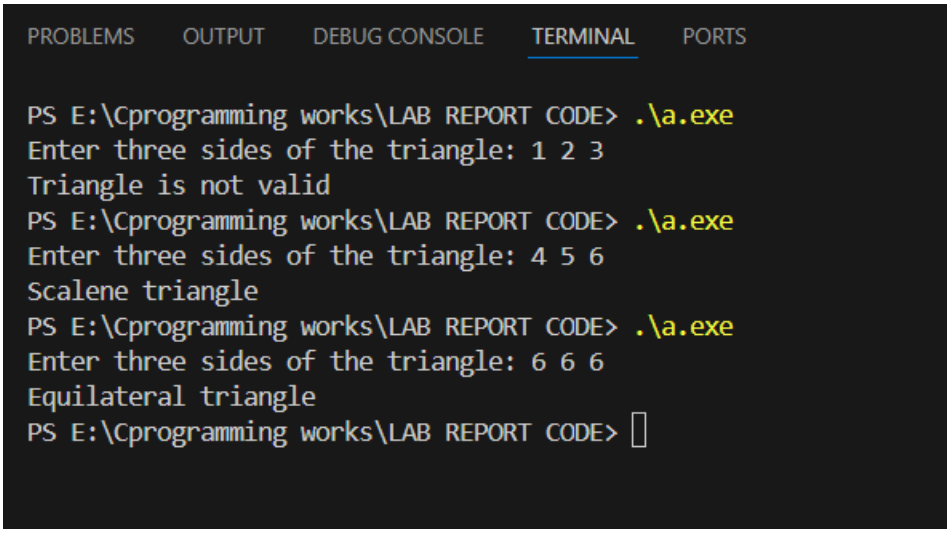
CODE :

```
        #include <stdio.h>

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

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

// Check if triangle is valid
if ((a + b > c) && (a + c > b) && (b + c > a)) {

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

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

   // Check Right-angled
   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");
   }
```

```
        // Otherwise, Scalene
        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

PSEUDOCODE:

START

Declare float weight, height, bmi

Print "Enter weight (kg): "

Input weight

Print "Enter height (m): "

Input height

*bmi = weight / (height * height)*

// BMI Category Check

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"

ELSE IF bmi >= 25 AND bmi <= 25.9 THEN
    Print "Overweight"

ELSE IF bmi >= 30 AND bmi <= 39.9 THEN
    Print "Obese"

ELSE
    Print "Invalid BMI value"

ENDIF

END
```

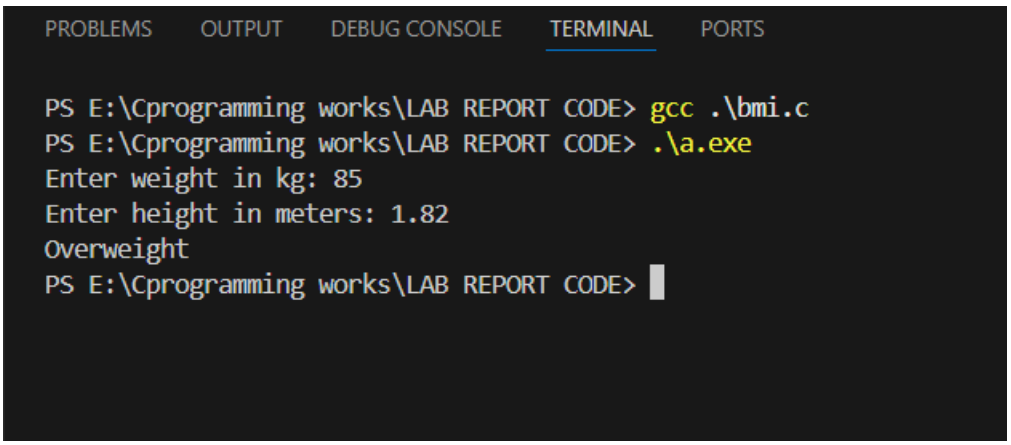
CODE :

```
#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);

// Calculate BMI
bmi = weight / (height * height);

// Check BMI Category
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\n");
}
```

```
    else if (bmi >= 25 && bmi <= 25.9) {
        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

PSEUDOCODE:

START

Declare float x1, y1, x2, y2, x3, y3, area

Print "Enter coordinates x1 y1:"

Input x1, y1

Print "Enter coordinates x2 y2:"

Input x2, y2

Print "Enter coordinates x3 y3:"

Input x3, y3

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

IF area == 0 THEN

Print "Points are Collinear"

ELSE

Print "Points are Not Collinear"

ENDIF

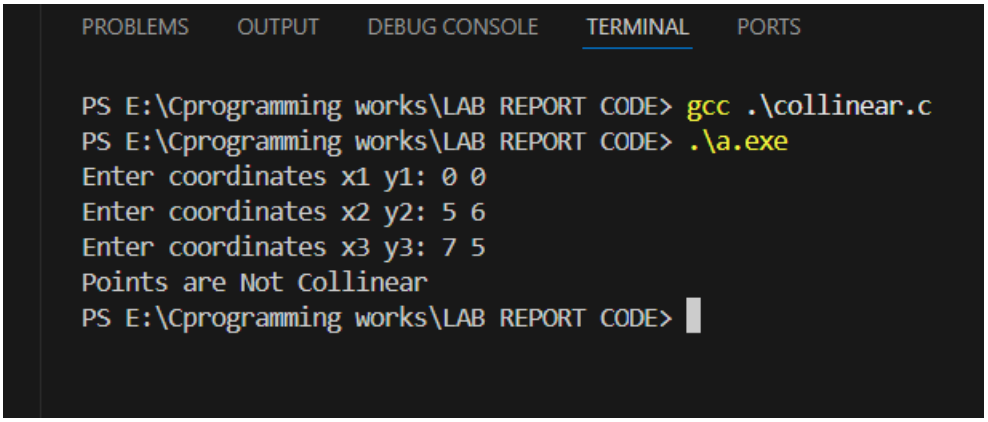
END

CODE :

```
#include <stdio.h>

int main() {
   float x1, y1, x2, y2, x3, y3, area;

   // Input coordinates
   printf("Enter coordinates x1 y1: ");
   scanf("%f %f", &x1, &y1);

   printf("Enter coordinates x2 y2: ");
   scanf("%f %f", &x2, &y2);
```

```
    printf("Enter coordinates x3 y3: ");
    scanf("%f %f", &x3, &y3);

    // Calculate area of the triangle formed by the points
    area = 0.5 * ( x1 * (y2 - y3) +
                x2 * (y3 - y1) +
                x3 * (y1 - y2) );

    // Check collinearity
    if (area == 0) {
        printf("Points are Collinear\n");
    } else {
        printf("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

PSEUDOCODE:

```
START

Declare integer year, i, total_days, day
Set total_days = 0

Print "Enter 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
```

```
    ENDIF
END FOR

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 "Invalid day"
ENDIF
```

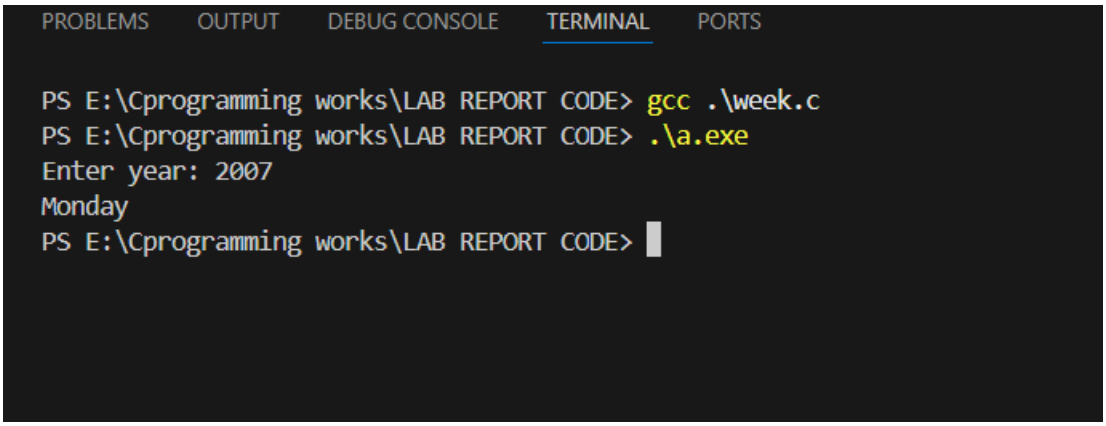
END

CODE :

```
#include <stdio.h>

int main() {
    int year, i, total_days = 0, day;

    // Input year
    printf("Enter year: ");
    scanf("%d", &year);

    // Count days from year 1 to (year - 1)
    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
        }
    }

    // Calculate day of week
```

```
    day = total_days % 7;

    // Print corresponding day
    if (day == 0)
        printf("Monday\n");
    else if (day == 1)
        printf("Tuesday\n");
    else if (day == 2)
        printf("Wednesday\n");
    else if (day == 3)
        printf("Thursday\n");
    else if (day == 4)
        printf("Friday\n");
    else if (day == 5)
        printf("Saturday\n");
    else if (day == 6)
        printf("Sunday\n");
    else
        printf("Invalid day\n");

    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

PSEUDOCODE:

START

Declare float l1, b1, l2, b2, l3, b3

Declare float p1, p2, p3, max_perimeter

Print "Enter l1 b1:"

Input l1, b1

Print "Enter l2 b2:"

Input l2, b2

Print "Enter l3 b3:"

Input l3, b3

*p1 = 2 * (l1 + b1)*

*p2 = 2 * (l2 + b2)*

*p3 = 2 * (l3 + b3)*

max_perimeter = (p1 > p2) ?

((p1 > p3) ? p1 : p3) :

((p2 > p3) ? p2 : p3)

Print "Maximum Perimeter =", max_perimeter

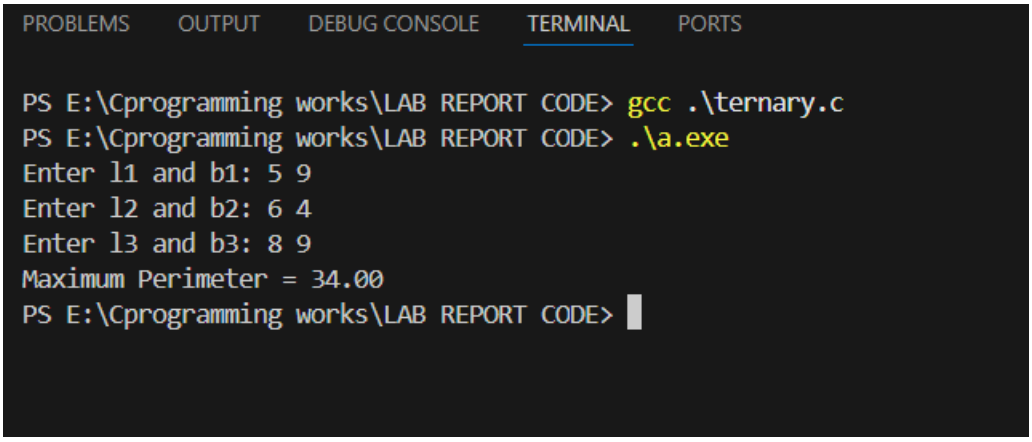
END

CODE:

```
#include <stdio.h>

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

   // Input lengths and breadths
   printf("Enter l1 and b1: ");
   scanf("%f %f", &l1, &b1);

   printf("Enter l2 and b2: ");
   scanf("%f %f", &l2, &b2);

   printf("Enter l3 and b3: ");
   scanf("%f %f", &l3, &b3);

   // Calculate perimeters
   p1 = 2 * (l1 + b1);
   p2 = 2 * (l2 + b2);
```

```
    p3 = 2 * (l3 + b3);

    // Find maximum perimeter using ternary operator
    max_perimeter = (p1 > p2) ?
                    ((p1 > p3) ? p1 : p3) :
                    ((p2 > p3) ? p2 : p3);

    printf("Maximum Perimeter = %.2f\n", max_perimeter);

    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

PSEUDOCODE :

START

Initialize pos = 0, neg = 0, zero = 0

Read num

IF num > 0 THEN

pos = pos + 1

ELSE

IF num < 0 THEN

neg = neg + 1

ELSE

zero = zero + 1

ENDIF

ENDIF

Read choice

WHILE choice == 1 DO

Read num

IF num > 0 THEN

pos = pos + 1

```
    ELSE
        IF num < 0 THEN
            neg = neg + 1
        ELSE
            zero = zero + 1
        ENDIF
    ENDIF

    Read choice
END WHILE

Print pos, neg, zero

END
```

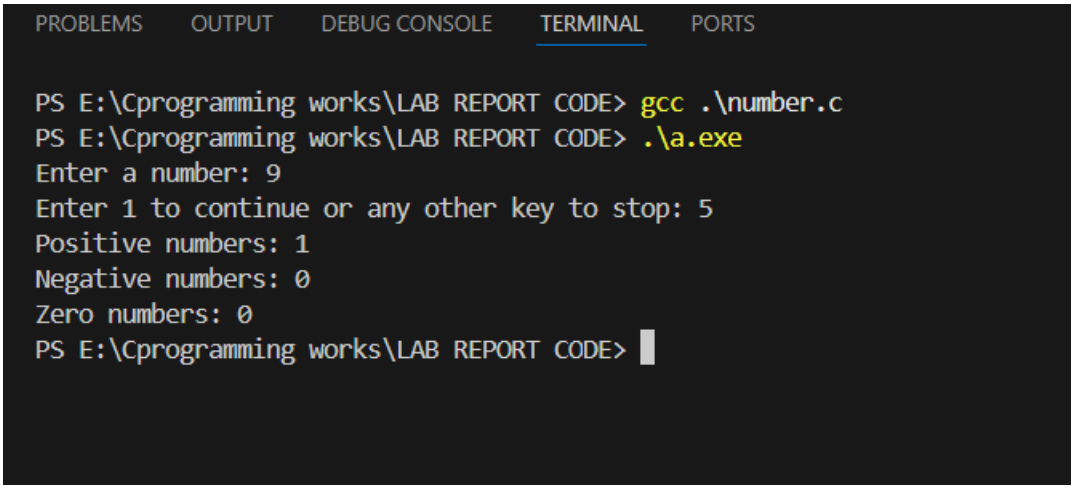
CODE :

```
#include <stdio.h>

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

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

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

printf("Enter 1 to continue or any other key to stop: ");
scanf("%d", &choice);

while (choice == 1) {
   printf("Enter a number: ");
   scanf("%d", &num);

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

   printf("Enter 1 to continue or any other key to stop: ");
```

```
    scanf("%d", &choice);
  }

  printf("Positive numbers: %d\n", pos);
  printf("Negative numbers: %d\n", neg);
  printf("Zero numbers: %d\n", zero);

  return 0;
}
```

OUTPUT :



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

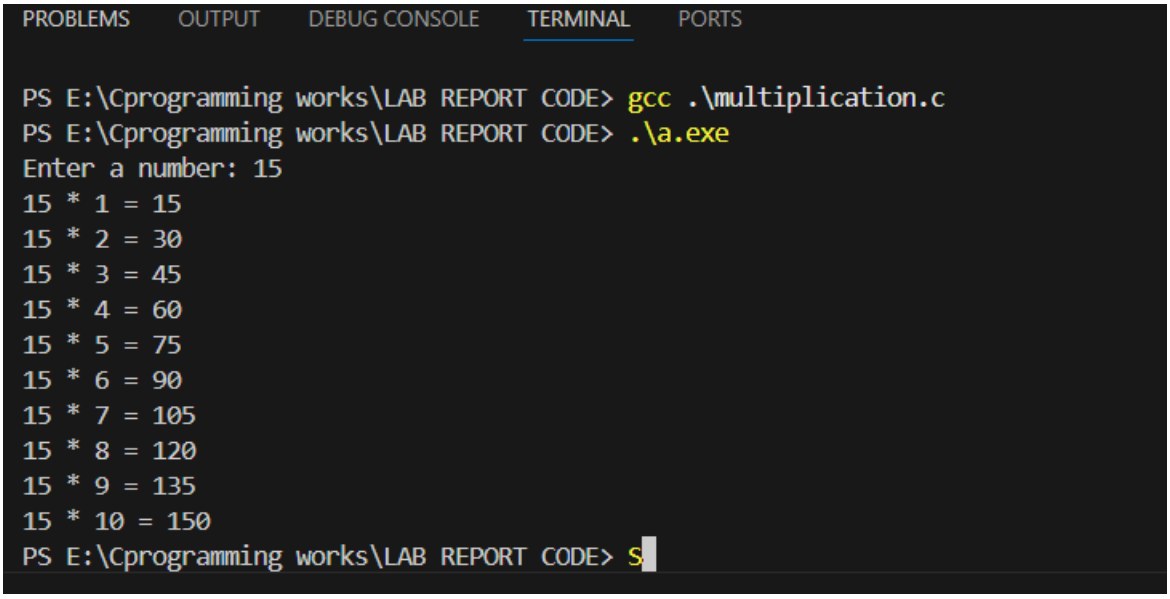
PSEUDOCODE:

```
START

Read num
Set i = 1

WHILE i <= 10 DO
   result = num * i
   Print num, "*", i, "=", result
   i = i + 1
END WHILE

END
```

CODE :

```
#include <stdio.h>

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

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

   i = 1;

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

   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

PSEUDOCODE :

```
START

Set i = 1
Set num = 1

WHILE i <= 3 DO
   space = 3

   WHILE space > i DO
      Print " "
      space = space - 1
   END WHILE

   j = 1
```

```
    WHILE j <= i DO
        Print num
        num = num + 1
        j = j + 1
    END WHILE

    Print newline

    i = i + 1
END WHILE

STOP
```

CODE :

```
#include <stdio.h>

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

    while (i <= 3) {
        space = 3;

        while (space > i) {
```

```
            printf(" ");
            space--;
        }

        j = 1;

        while (j <= i) {
            printf("%d", num);
            num++;
            j++;
        }

        printf("\n");
        i++;
    }

    return 0;
}
```

OUTPUT:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\Cprogramming works\LAB REPORT CODE> gcc .\pattern.c
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe
  1
 23
456
PS E:\Cprogramming works\LAB REPORT CODE>
```

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

PSEUDOCODE :

```
START

Set n = 4
Set i = 0

WHILE i < n DO
   space = 1

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

   coef = 1
   j = 0

   WHILE j <= i DO
```

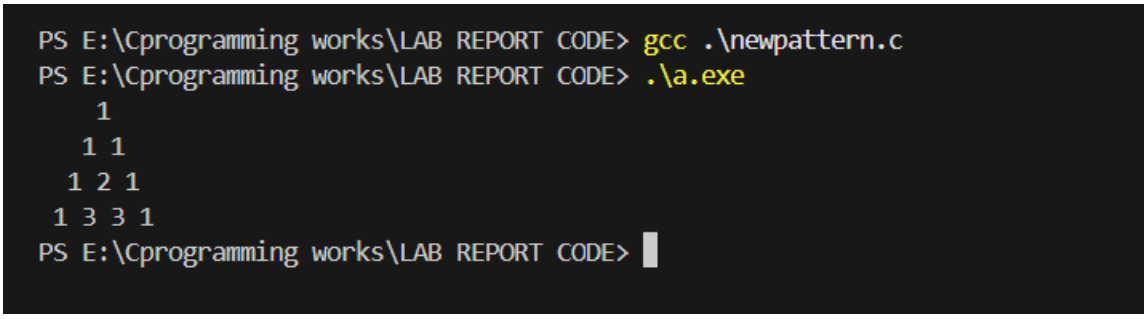
```
    Print coef
    coef = coef * (i - j) / (j + 1)
    j = j + 1
  END WHILE

  Print newline

  i = i + 1
END WHILE

STOP
```

CODE :

```
#include <stdio.h>

int main() {
  int n = 4, i = 0, j, space;
  int coef;

  while (i < n) {
    space = 1;
    while (space <= n - i) {
      printf(" ");
      space++;
```

```
        }

        coef = 1;
        j = 0;

        while (j <= i) {
            printf("%d ", coef);
            coef = coef * (i - j) / (j + 1);
            j++;
        }

        printf("\n");
        i++;
    }

    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

PSEUDOCODE :

START

Set population = 100000

Print "Population of the town over the last 10 years:"

Set year = 10

```
WHILE year >= 1 DO
    population = population / 1.10
    Print year, population
    year = year - 1
END WHILE

STOP
```

CODE :

```
#include <stdio.h>

int main() {
    float population = 100000;
    int year = 10;

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

    while (year >= 1) {
        population = population / 1.10;
        printf("Year %d: %.2f\n", year, population);
        year--;
    }

    return 0;
}
```

OUTPUT :

```
PS E:\Cprogramming works\LAB REPORT CODE> gcc .\population.c
PS E:\Cprogramming works\LAB REPORT CODE> .\a.exe
Population of the town over the last 10 years:
Year 10: 90909.09
Year 9: 82644.63
Year 8: 75131.48
Year 7: 68301.35
Year 6: 62092.14
Year 5: 56447.40
Year 4: 51315.82
Year 3: 46650.74
Year 2: 42409.77
Year 1: 38554.33
PS E:\Cprogramming works\LAB REPORT CODE>
```