Ashutosh Shrivastava

SAP id:590027859

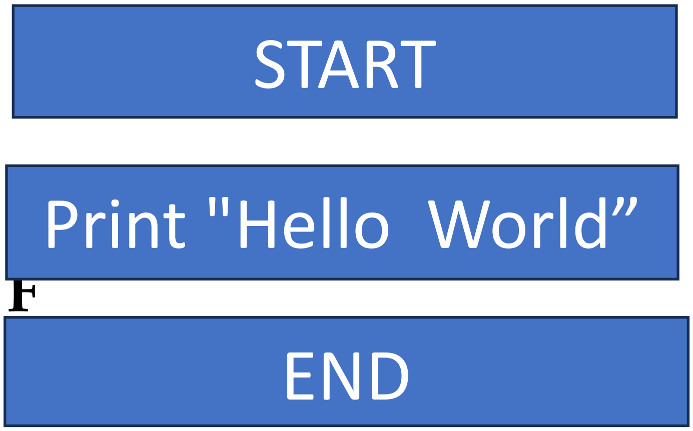
C Lab Report

List of Experiments

Exp 1: Installation,Environment Setup and starting with C language

1. **Write a C program to print “HelloWorld”**

Flowchart:



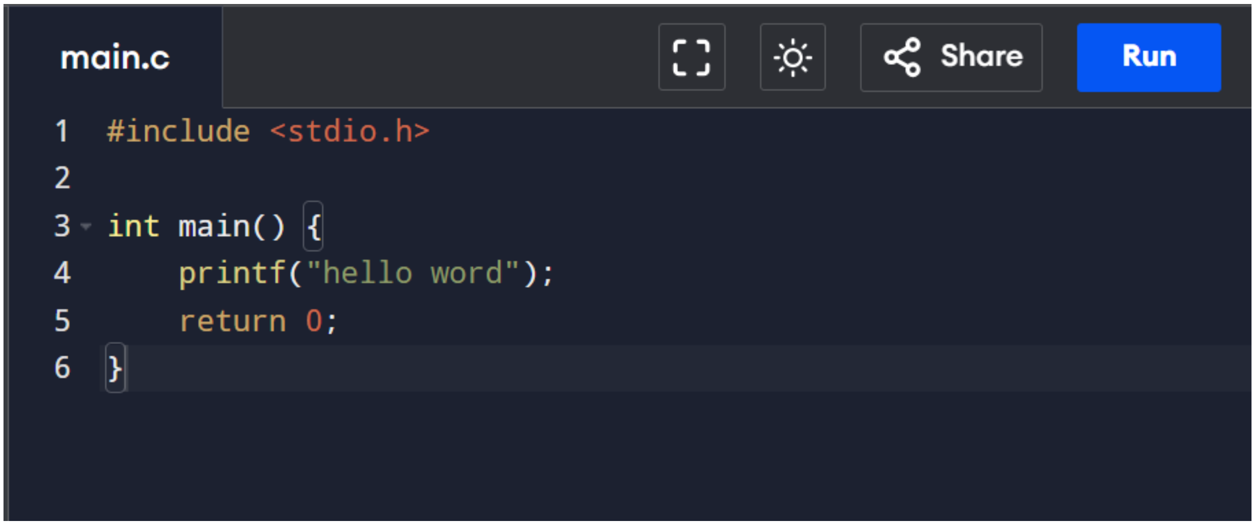
ALGORITHIM:

Step 1: Start the program
Step 2: Display the message "Hello World" on the screen
Step 3: End the program

Pseudo code:

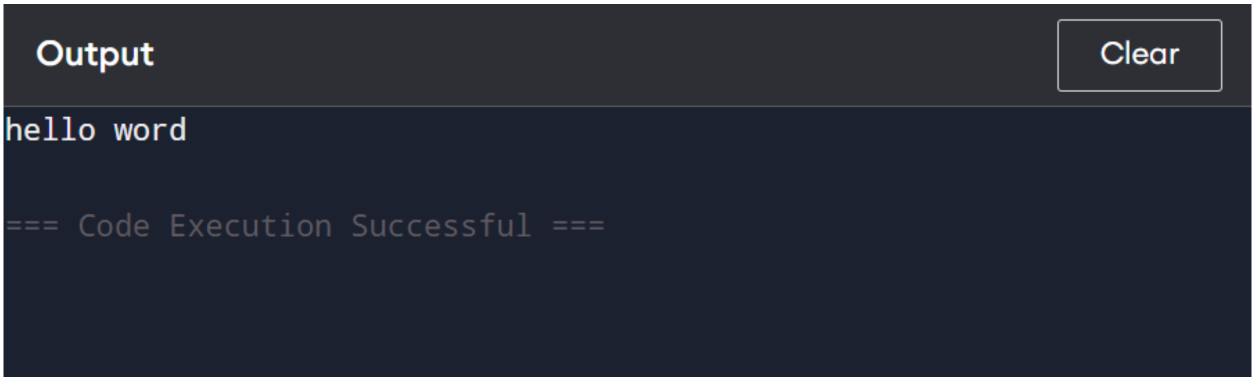
```
START
   PRINT "Hello World"
END
```

Code:



```c
#include <stdio.h>

int main() {
    printf("hello word");
    return 0;
}
```

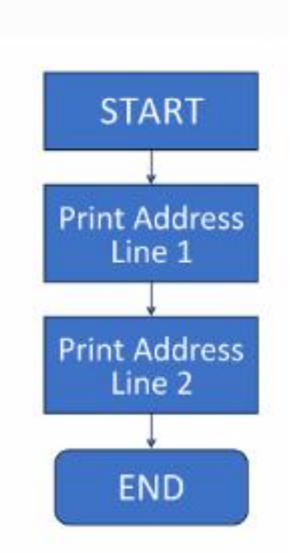
Result:



```
hello word

=== Code Execution Successful ===
```

2. Write a C Program to print the address in multiple lines(newline).

Flowchart:



Algorithim:

Step 1: Start the program

Step 2: Print the first line of the address

Step 3: Print the second line of the address

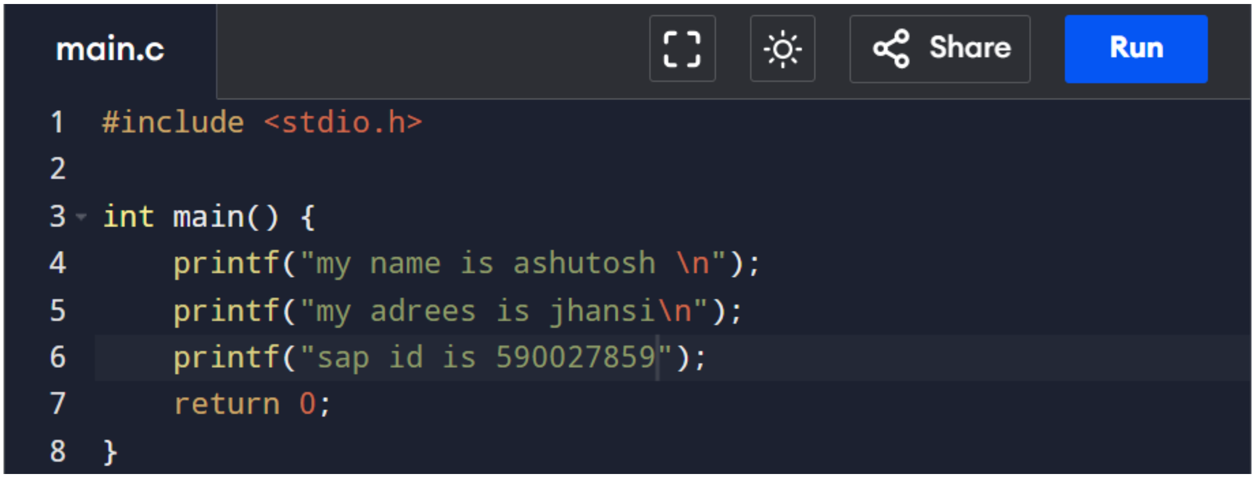
Step 4: Print the third line of the address

Step 5: End the program

Pseudo code:

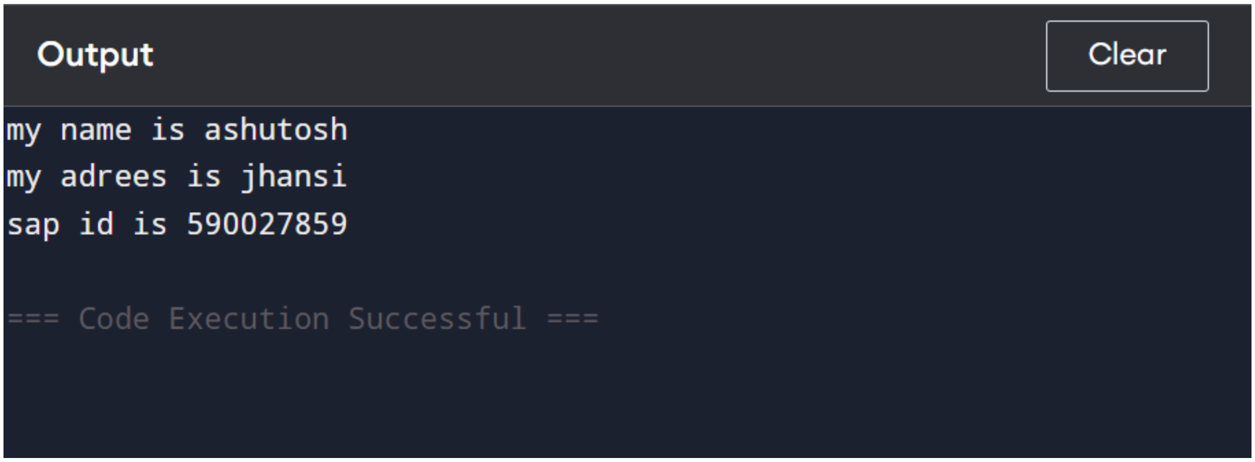
```
START
   PRINT "Your Address Line 1"
   PRINT "Your Address Line 2"
   PRINT "Your Address Line 3"
END
```

Code:



```c
#include <stdio.h>

int main() {
    printf("my name is ashutosh \n");
    printf("my adrees is jhansi\n");
    printf("sap id is 590027859");
    return 0;
}
```

Result



```
my name is ashutosh
my adrees is jhansi
sap id is 590027859


=== Code Execution Successful ===
```

3.write a program that prompts the user to enter their name and age.

Algorithim:

Step 1: Start the program

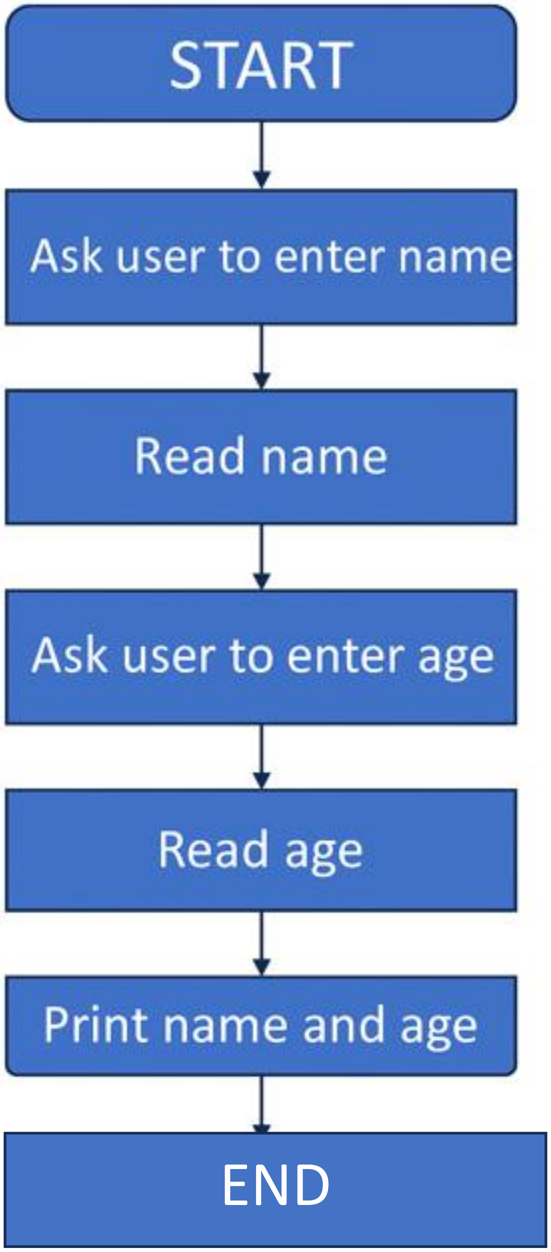
Step 2: Display a message asking the user to

enter their name
Step 3: Accept the user's name
Step 4: Display a message asking the user to enter their age
Step 5: Accept the user's age
Step 6: Display the entered name and age
Step 7: End the program

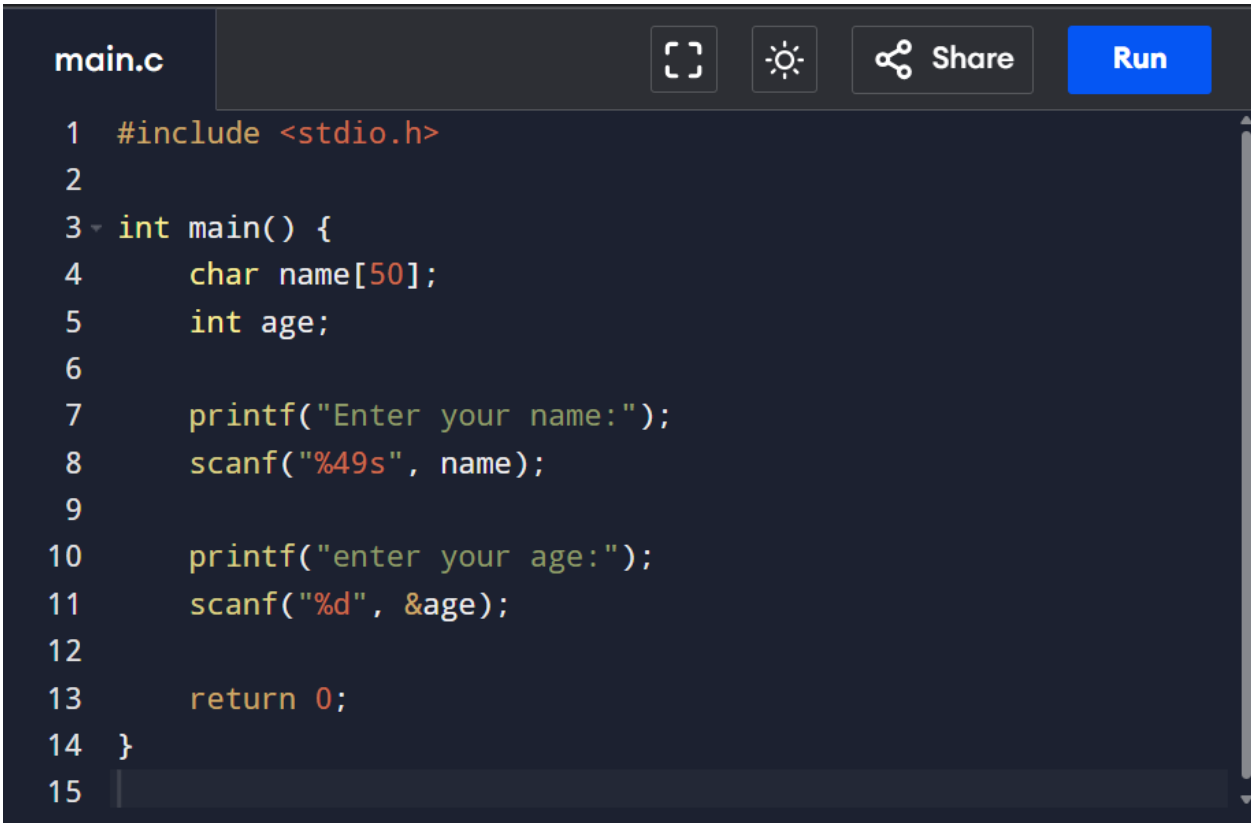
Pseudo code:

```
START
   ASK user to enter name
   READ name
   ASK user to enter age
   READ age
   PRINT name and age
END
```

Flowchart

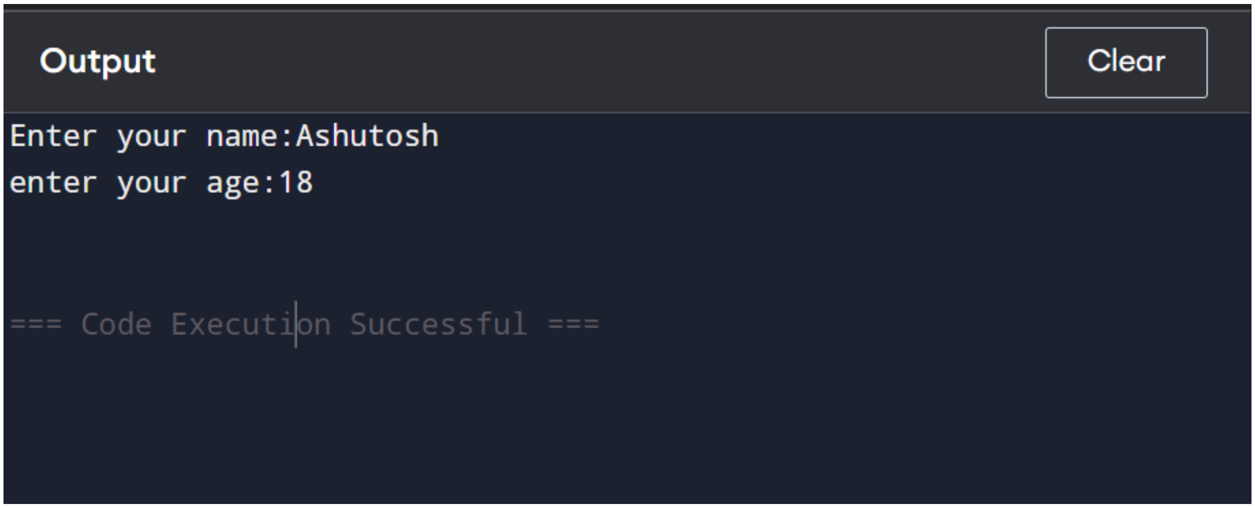

START
Ask user to enter name
Read name
Ask user to enter age
Read age
Print name and age
END

Code:



```c
#include <stdio.h>

int main() {
    char name[50];
    int age;

    printf("Enter your name:");
    scanf("%49s", name);

    printf("enter your age:");
    scanf("%d", &age);

    return 0;
}

```

Result



```
Enter your name:Ashutosh
enter your age:18


=== Code Execution Successful ===
```

4. Write a C program to add two numbers, take number from user.

Algorithim:

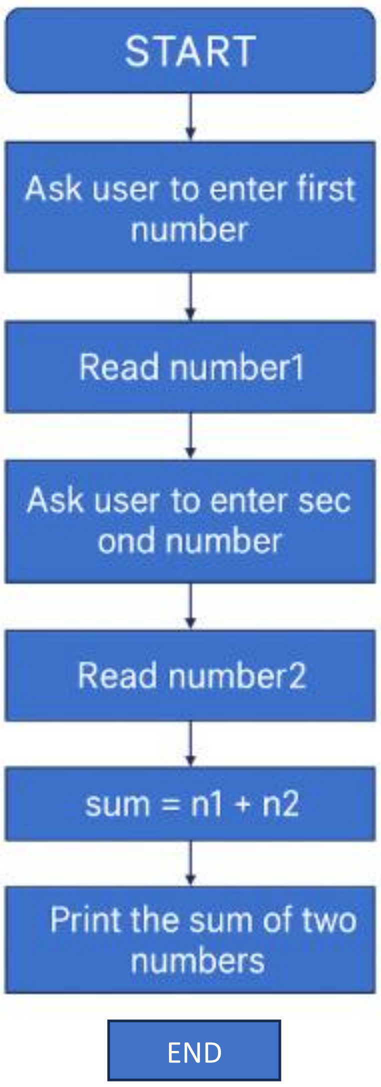
- □ Start
- □ Declare variables a, b, and sum.
- □ Read the first number and store it in a.
- □ Read the second number and store it in b.
- □ Calculate sum = a + b.
- □ Display the value of sum.
- □ End

Pseudo code:

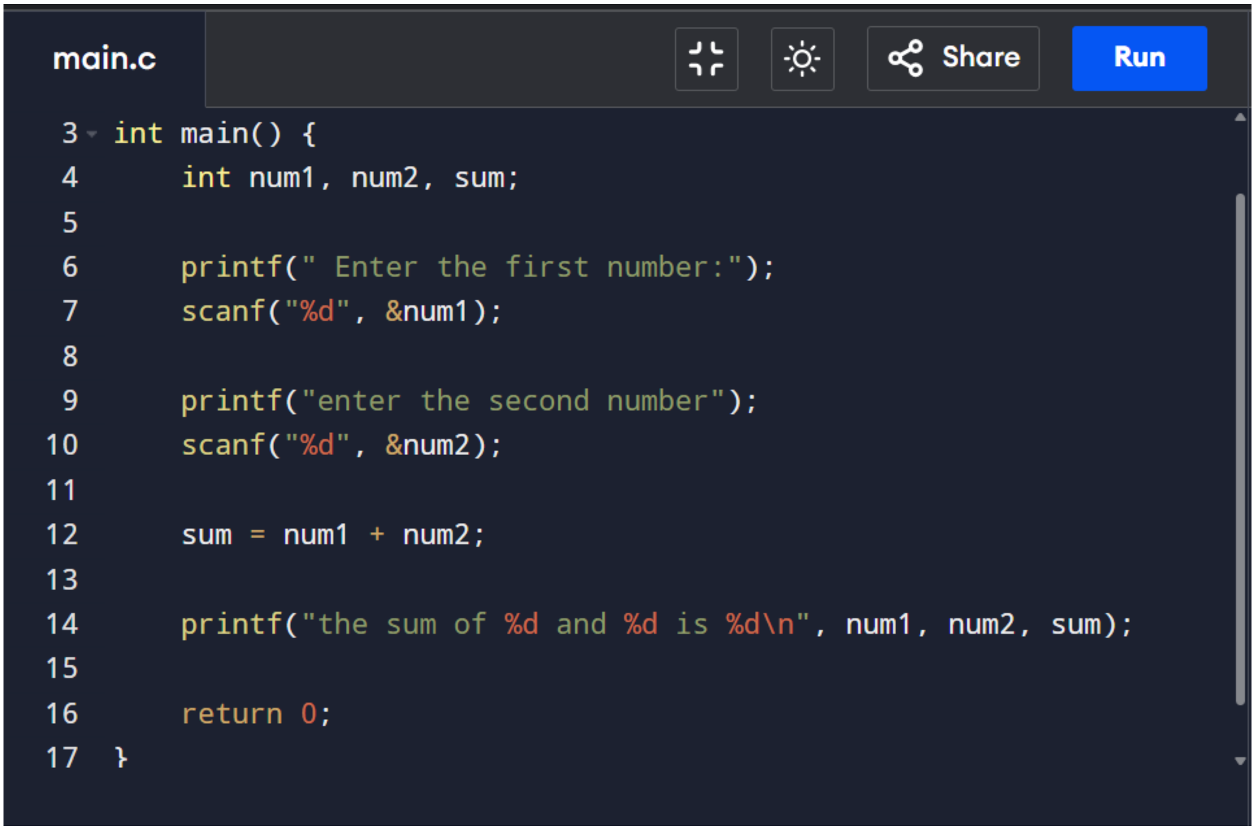
```
START
   DECLARE a, b, sum

   PRINT "Enter first number:"
   INPUT a

   PRINT "Enter second number:"
   INPUT b

   sum ← a + b
```

```
    PRINT "The sum is:", sum
END
```

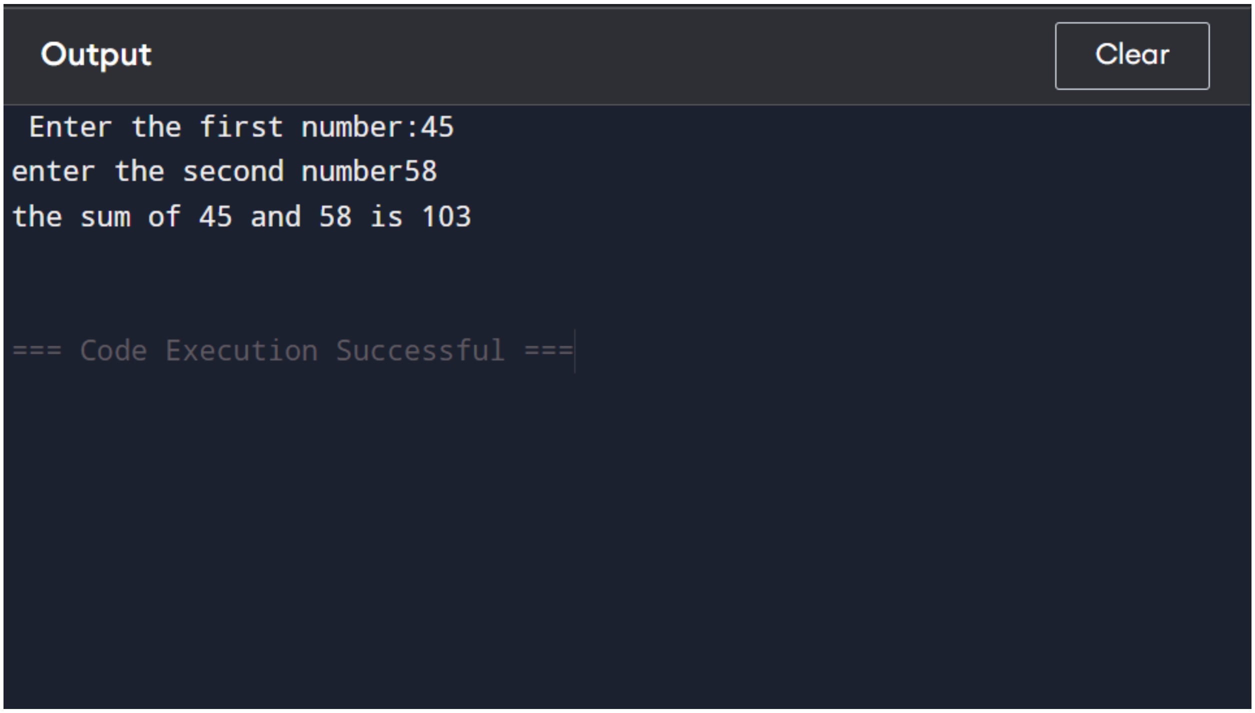
Flowchart:


START
Ask user to enter first number
Read number1
Ask user to enter sec ond number
Read number2
sum = n1 + n2
Print the sum of two numbers
END

Code:



```c
int main() {
    int num1, num2, sum;

    printf(" Enter the first number:");
    scanf("%d", &num1);

    printf("enter the second number");
    scanf("%d", &num2);

    sum = num1 + num2;

    printf("the sum of %d and %d is %d\n", num1, num2, sum);

    return 0;
}
```



```
 Enter the first number:45
enter the second number58
the sum of 45 and 58 is 103


=== Code Execution Successful ===
```

3. Opetators:

1. WAP a C program calculate the area and perimeter of rectangle based on its length and width.

Algorithim:

□ Start

□ Declare variables: length, width, area, perimeter.

□ Ask user to enter the length.

□ Read the length.

□ Ask user to enter the width.

□ Read the width.

□ Calculate area → area = length × width

□ Calculate perimeter → perimeter = 2 × (length + width)

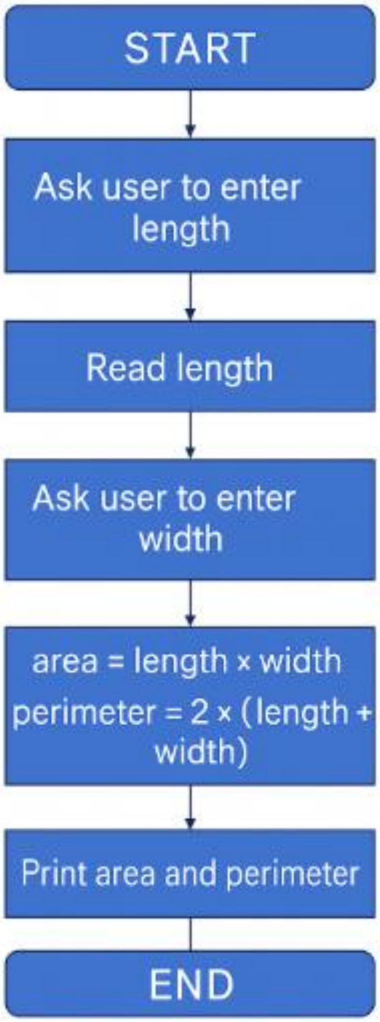
□ Display area and perimeter.

□ End

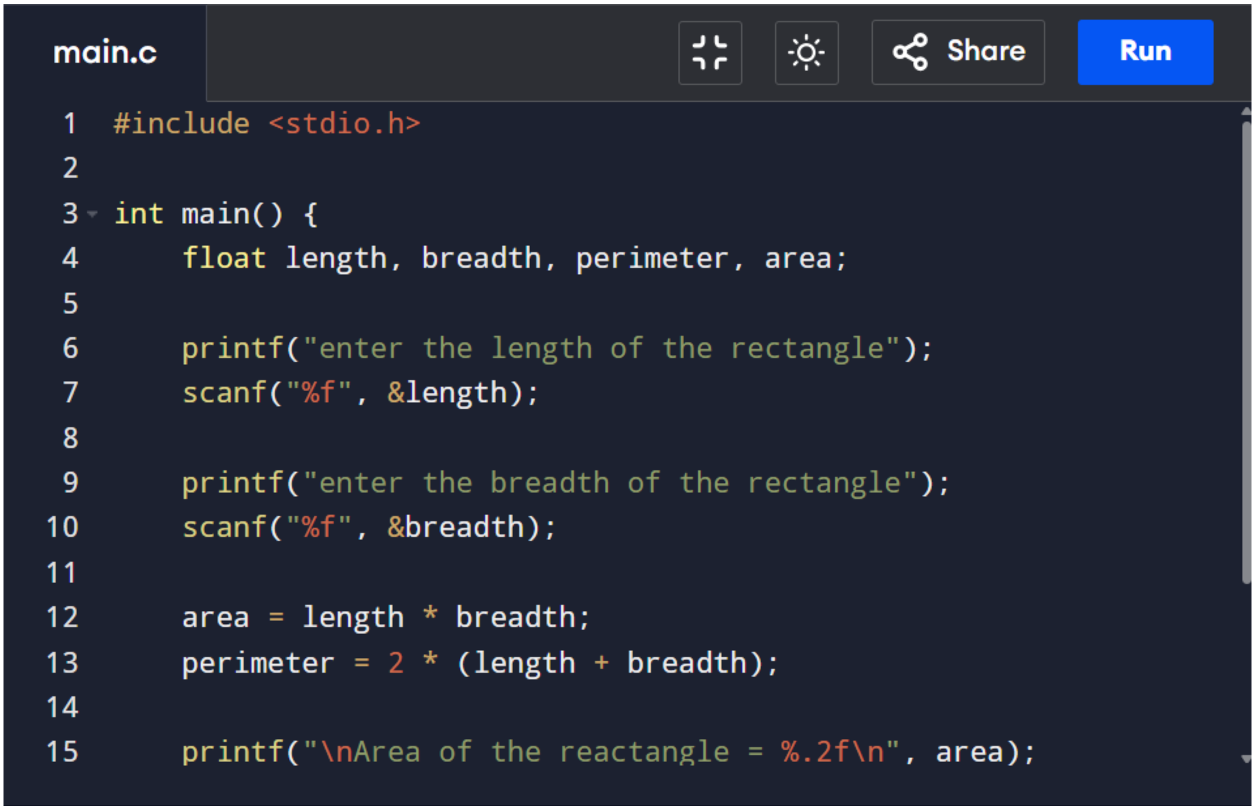
Pseudo code:

```
START

DECLARE length, width, area, perimeter

PRINT "Enter length:"
INPUT length

PRINT "Enter width:"
INPUT width

area ← length * width
perimeter ← 2 * (length + width)

PRINT "Area =", area
PRINT "Perimeter =", perimeter

END
```

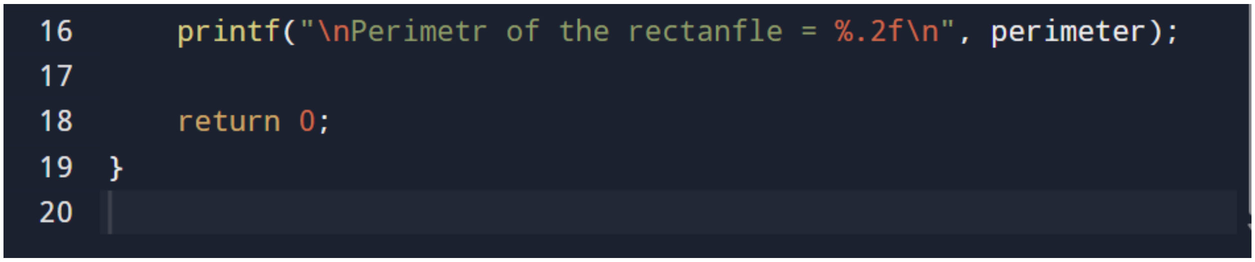
Flowchart:


START
Ask user to enter length
Read length
Ask user to enter width
area = length × width
perimeter = 2 × (length + width)
Print area and perimeter
END

Code:

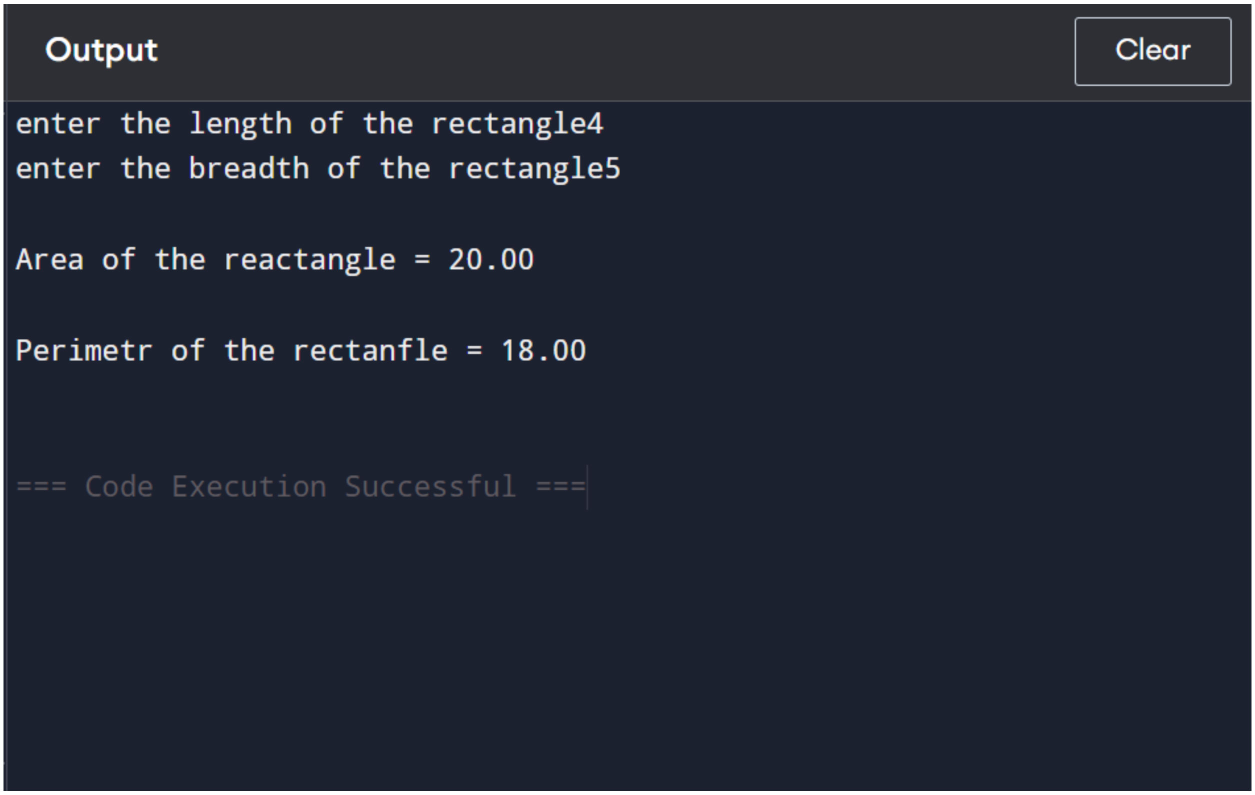


```c
#include <stdio.h>

int main() {
    float length, breadth, perimeter, area;

    printf("enter the length of the rectangle");
    scanf("%f", &length);

    printf("enter the breadth of the rectangle");
    scanf("%f", &breadth);

    area = length * breadth;
    perimeter = 2 * (length + breadth);

    printf("\nArea of the reactangle = %.2f\n", area);
```



```c
    printf("\nPerimetr of the rectanfle = %.2f\n", perimeter);

    return 0;
}

```

result



```
enter the length of the rectangle4
enter the breadth of the rectangle5

Area of the reactangle = 20.00

Perimetr of the rectanfle = 18.00



=== Code Execution Successful ===
```

2. 2. WAP a C program to Convert temperature from Celsius to Fahrenheit using the formula :F=(C*9/5)+32

Algorithim

1. Start
2. Declare variables: C (Celsius), F (Fahrenheit)
3. Ask user to enter temperature in Celsius

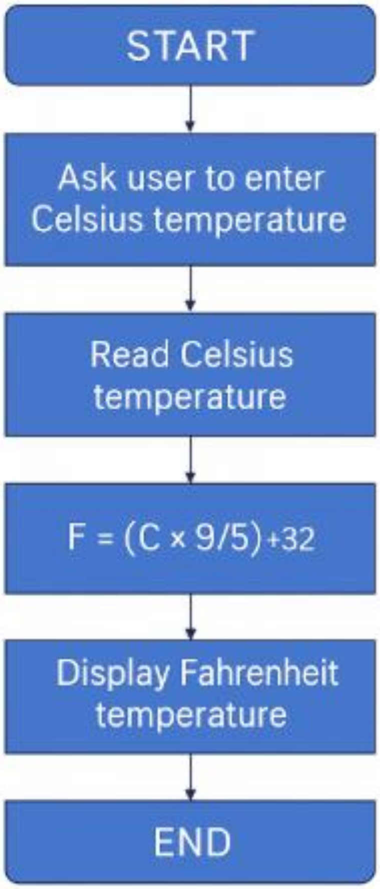
4. **Read the value of C**
5. **Calculate Fahrenheit using**
   $F = (C \times 9/5) + 32$
6. **Display the value of F**
7. **End**

Pseudo code:

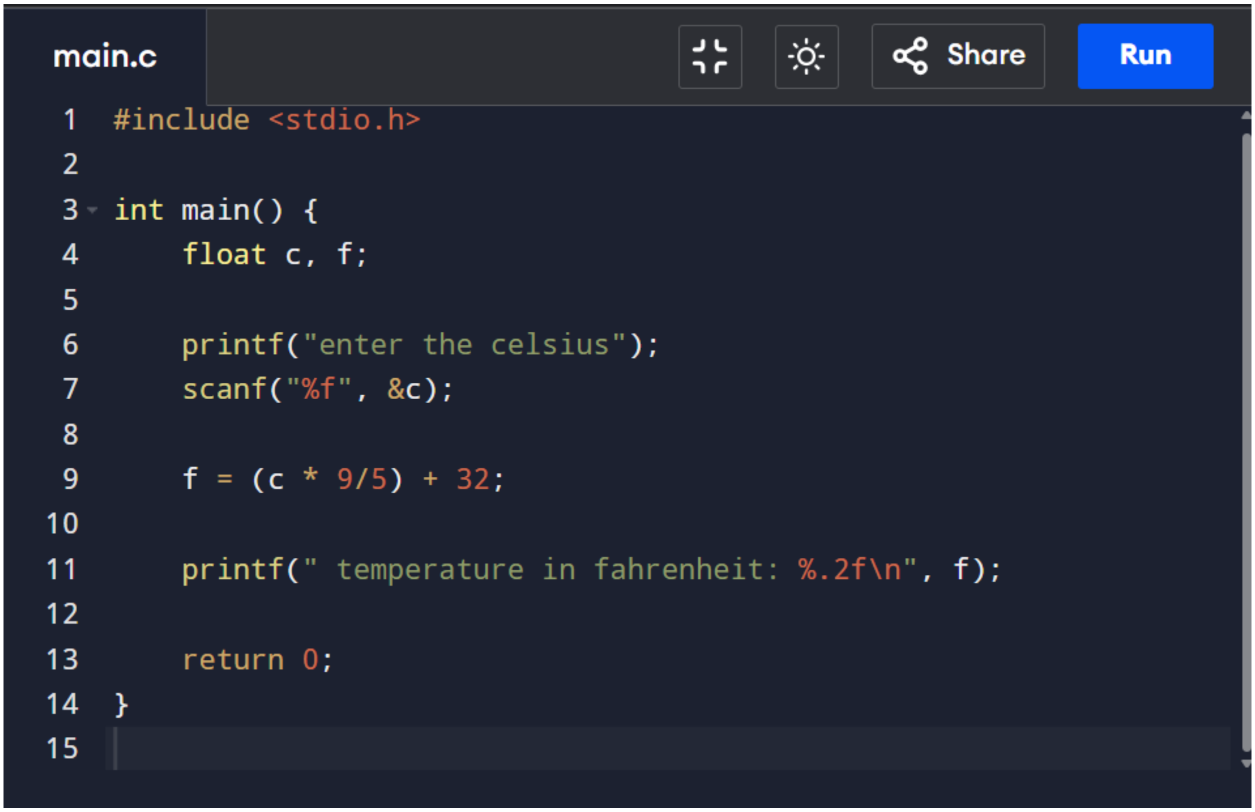
```
START

DECLARE C, F

PRINT "Enter temperature in Celsius:"
INPUT C

$F \leftarrow (C * 9/5) + 32$

PRINT "Temperature in Fahrenheit =", F
```

END

Flowchart:

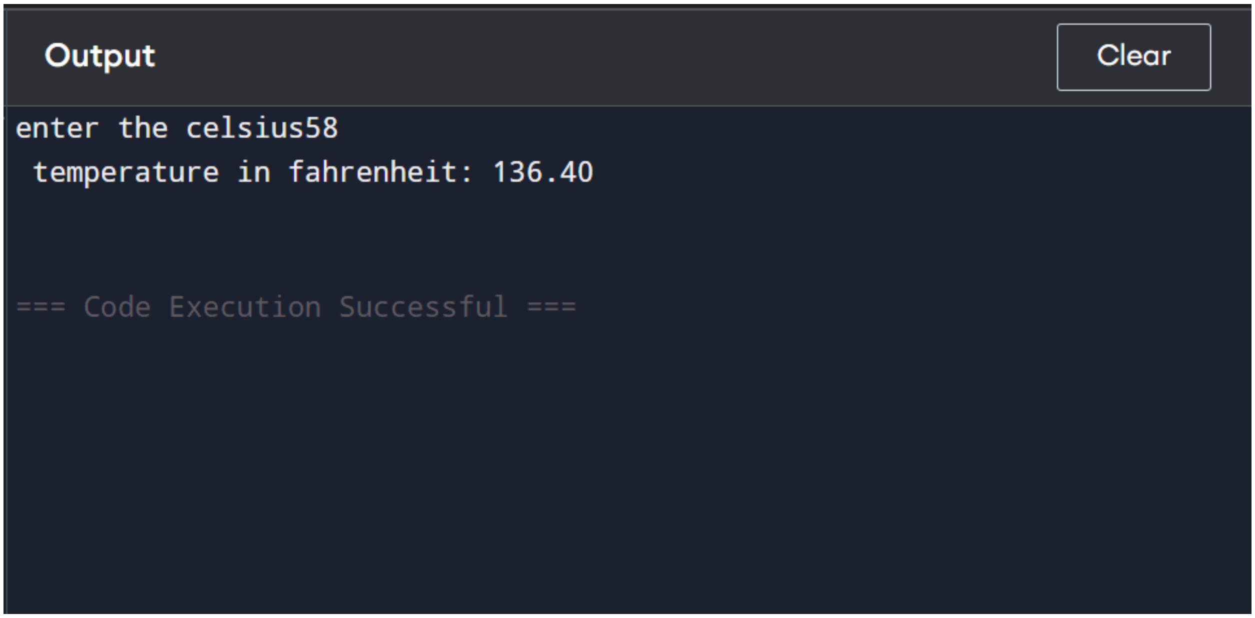

START
Ask user to enter Celsius temperature
Read Celsius temperature
F = (C x 9/5)+32
Display Fahrenheit temperature
END

Code:



```c
#include <stdio.h>

int main() {
    float c, f;

    printf("enter the celsius");
    scanf("%f", &c);

    f = (c * 9/5) + 32;

    printf(" temperature in fahrenheit: %.2f\n", f);

    return 0;
}

```

Result:



```
enter the celsius58
 temperature in fahrenheit: 136.40


=== Code Execution Successful ===
```

3.Conditional statements

1.1. WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles,equilateral, rightangle,orscalene.Take sides of the triangle as input from a user.

Algorithim

□ Start

□ Input the three sides: a, b, c.

□ Check validity of triangle using the rule:

- **A triangle is valid if a + b > c AND a + c > b AND b + c > a**

□ If the triangle is not valid,

- **Print "Triangle is not valid"**
- **Go to End**

□ If triangle is valid, check type:

- **If a == b == c → It is Equilateral**

- **Else if a == b OR b == c OR a == c → It is Isosceles**
- **Else if $a^2 + b^2 = c^2$ OR $a^2 + c^2 = b^2$ OR $b^2 + c^2 = a^2$ → Right-Angle Triangle**
- **Else → Scalene Triangle**

□ **Print the type of triangle.**

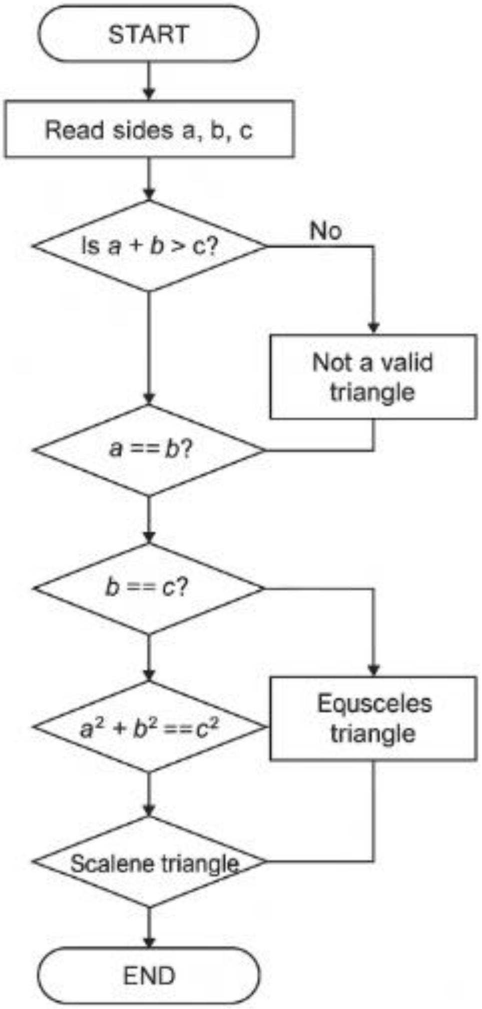
□ **End**

Pseudo code:

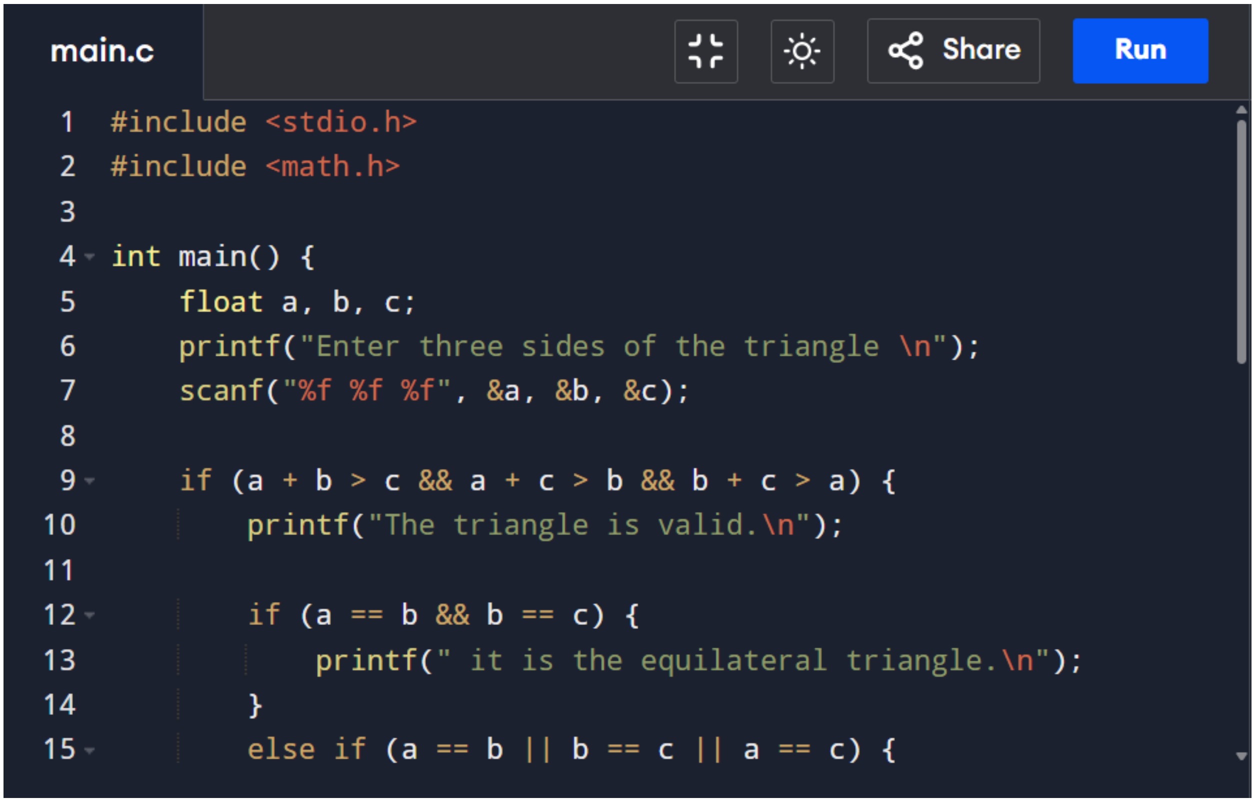
```
START

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
              a*a + c*c == b*b OR
              b*b + c*c == a*a) THEN
        PRINT "Right-Angle Triangle"
    ELSE
        PRINT "Scalene Triangle"
    ENDIF

ELSE
    PRINT "Triangle is not valid"
ENDIF

END
```

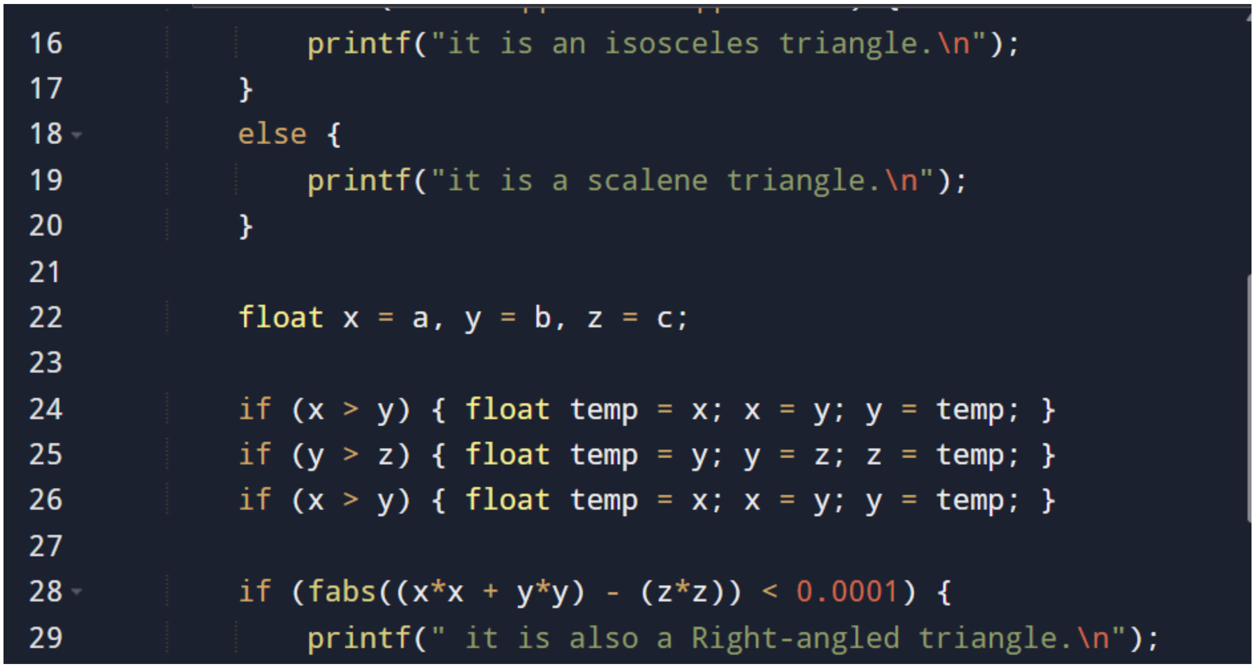
Flowchart:


START
Read sides a, b, c
Is $a + b > c$?
No
Not a valid triangle
$a == b$?
$b == c$?
$a^2 + b^2 == c^2$
Equsceles triangle
Scalene triangle
END

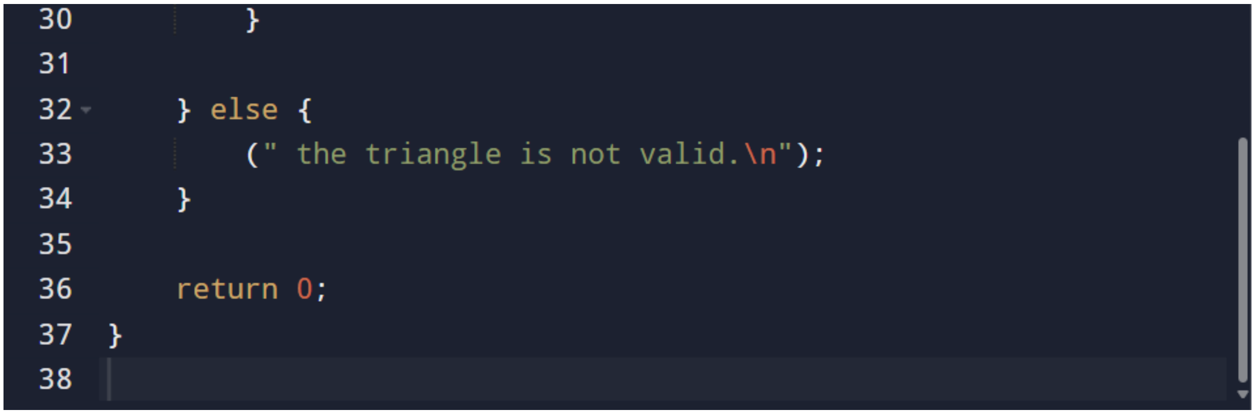
code:



```c
#include <stdio.h>
#include <math.h>

int main() {
    float a, b, c;
    printf("Enter three sides of the triangle \n");
    scanf("%f %f %f", &a, &b, &c);

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

        if (a == b && b == c) {
            printf(" it is the equilateral triangle.\n");
        }
        else if (a == b || b == c || a == c) {
```

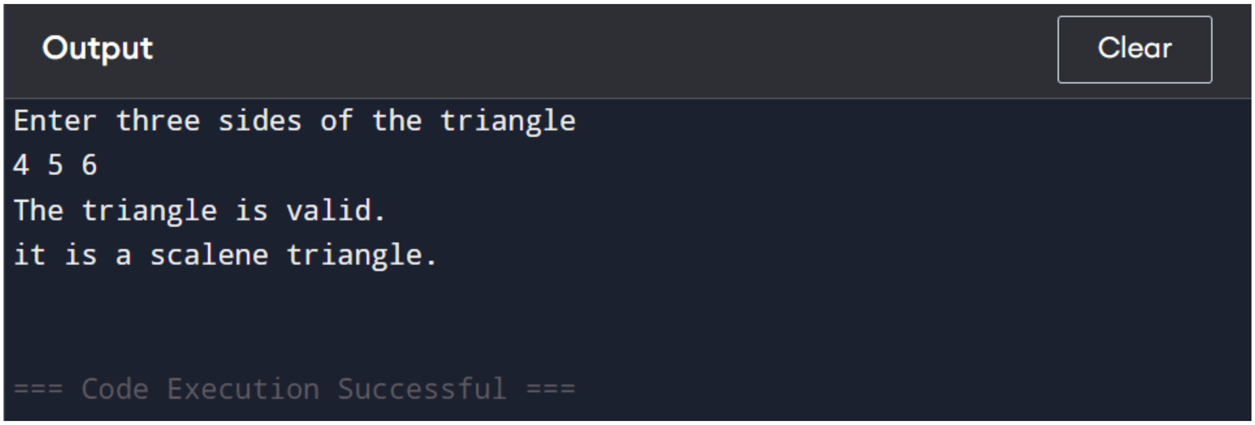


```c
            printf("it is an isosceles triangle.\n");
        }
        else {
            printf("it is a scalene triangle.\n");
        }

        float x = a, y = b, z = c;

        if (x > y) { float temp = x; x = y; y = temp; }
        if (y > z) { float temp = y; y = z; z = temp; }
        if (x > y) { float temp = x; x = y; y = temp; }

        if (fabs((x*x + y*y) - (z*z)) < 0.0001) {
            printf(" it is also a Right-angled triangle.\n");
```



```
        }

    } else {
        (" the triangle is not valid.\n");
    }

    return 0;
}

```

Result:



```
Enter three sides of the triangle
4 5 6
The triangle is valid.
it is a scalene triangle.


=== Code Execution Successful ===
```

2.2. WAPtocomputetheBMI Indexof thepersonandprint theBMIvaluesasper thefollowingranges.Youcanusethefollowingformulatocompute BMI= weight(kgs)/Height(Mts)*Height(Mts).

Algorithim

□ **Start**

□ **Input weight (kg) and height (m)**

□ **Calculate BMI = weight / (height × height)**

□ **Check BMI range:**

- **$BMI < 15$ → Starvation**
- **$15.1 \le BMI \le 17.5$ → Anorexic**
- **$17.6 \le BMI \le 18.5$ → Underweight**
- **$18.6 \le BMI \le 24.9$ → Ideal**
- **$25 \le BMI \le 29.9$ → Overweight**
- **$30 \le BMI \le 39.9$ → Obese**
- **$BMI \ge 40$ → Morbidly Obese**

□ **Print BMI and category**

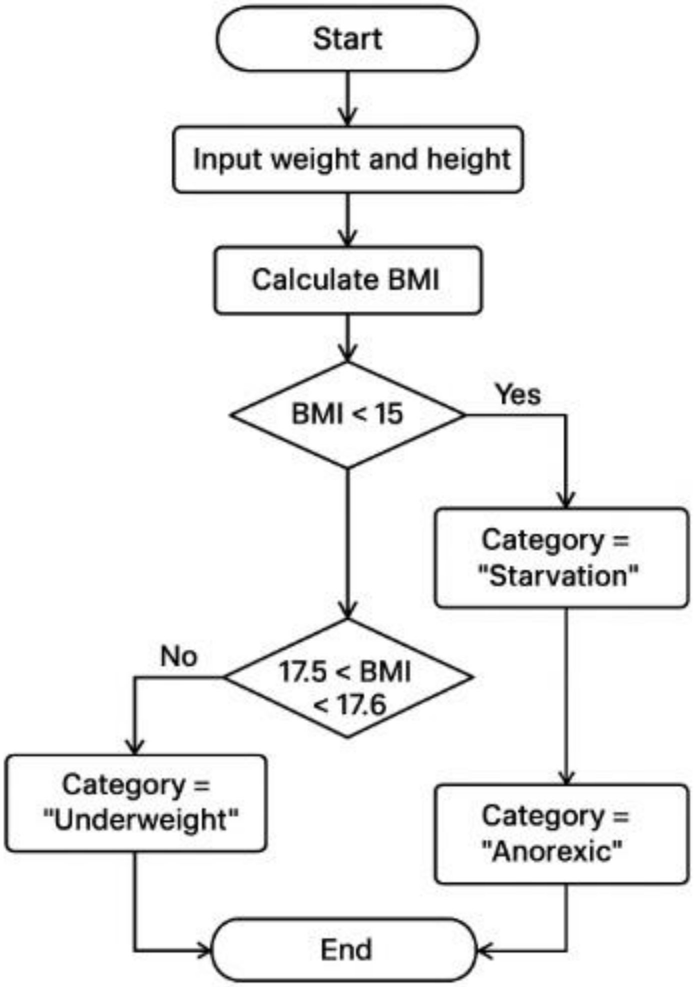
□ **End**

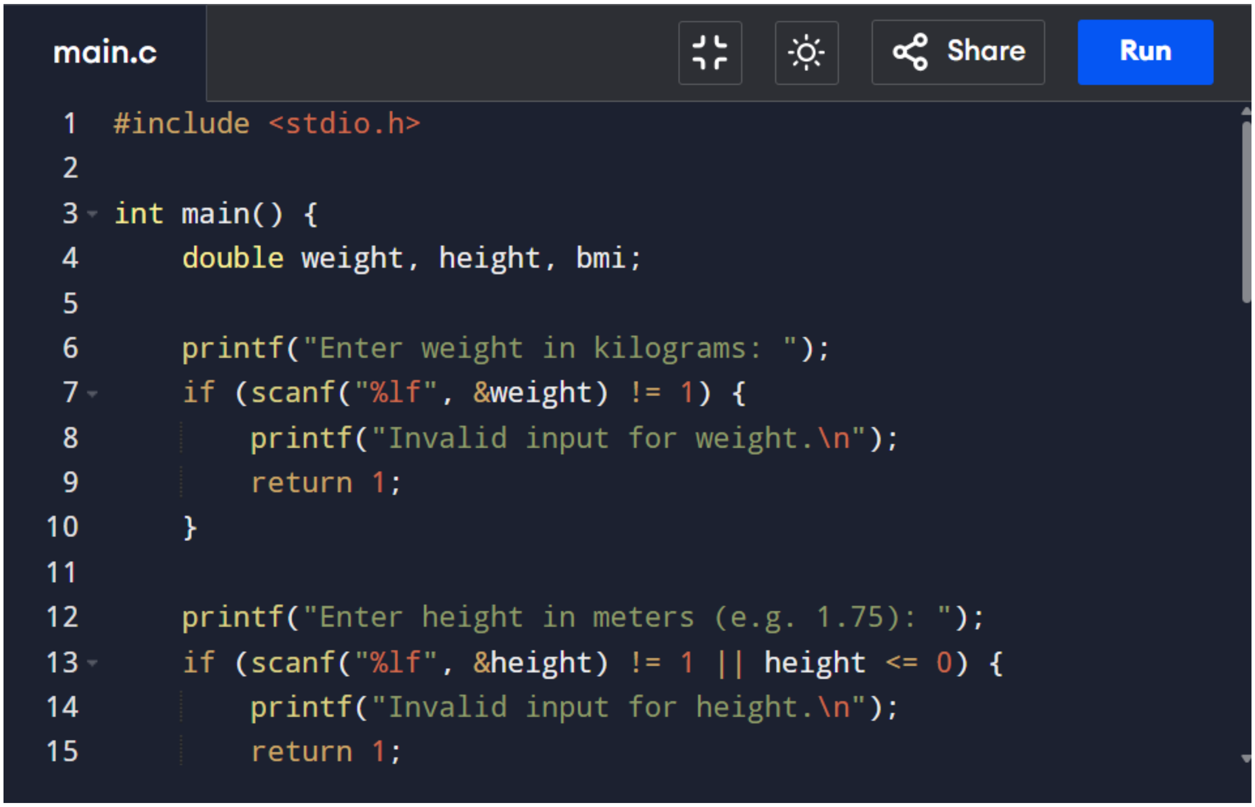
Pseudo code

```
START
  INPUT weight, height
  BMI = weight / (height * height)
```

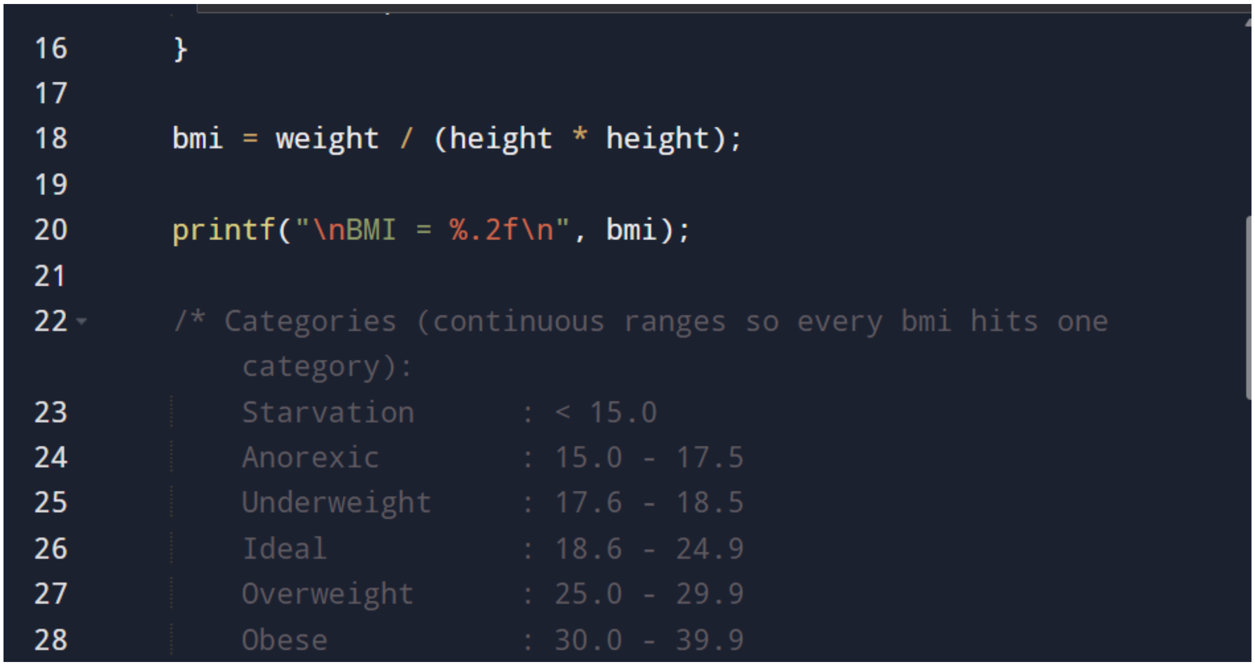
```
 IF BMI < 15 THEN
    PRINT "Starvation"
 ELSE IF BMI >= 15.1 AND BMI <= 17.5 THEN
    PRINT "Anorexic"
 ELSE IF BMI >= 17.6 AND BMI <= 18.5 THEN
    PRINT "Underweight"
 ELSE IF BMI >= 18.6 AND BMI <= 24.9 THEN
    PRINT "Ideal"
 ELSE IF BMI >= 25 AND BMI <= 29.9 THEN
    PRINT "Overweight"
 ELSE IF BMI >= 30 AND BMI <= 39.9 THEN
    PRINT "Obese"
 ELSE
    PRINT "Morbidity Obese"
 END IF
END
```

Flowchart

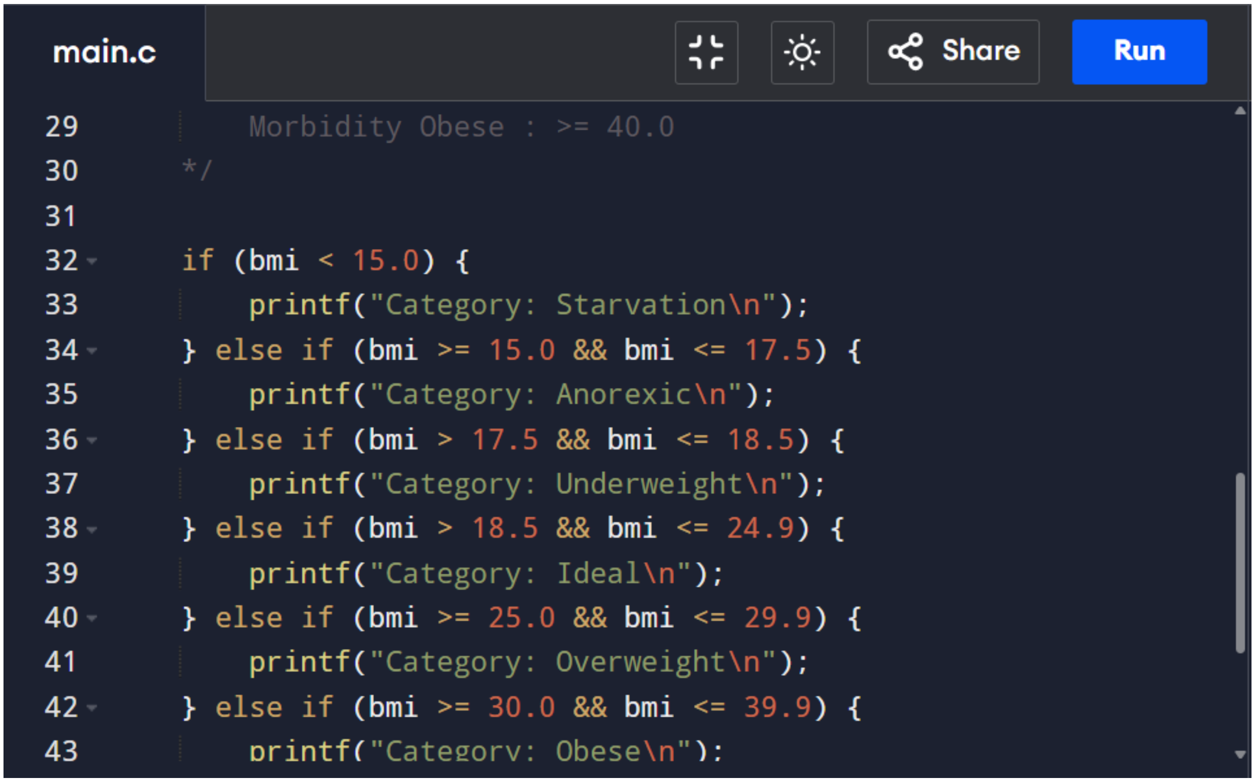

Start
Input weight and height
Calculate BMI
BMI < 15
Yes
Category = "Starvation"
No
17.5 < BMI < 17.6
Category = "Underweight"
Category = "Anorexic"
End



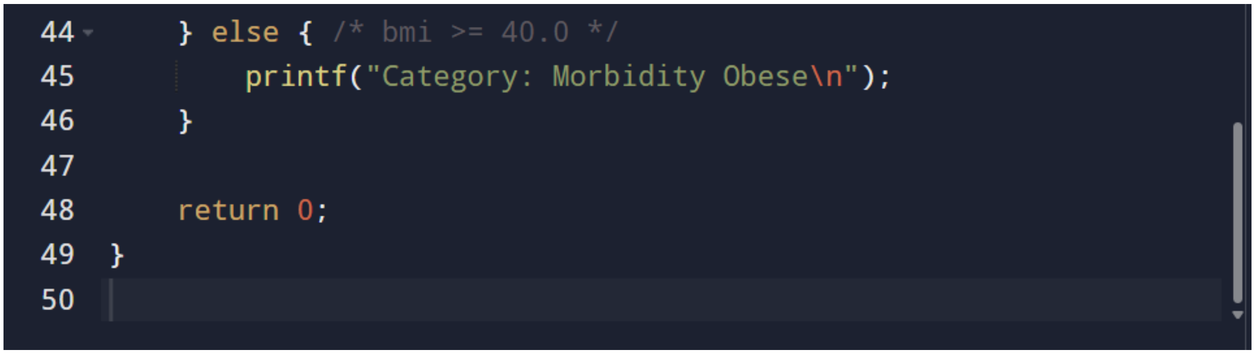
```c
#include <stdio.h>

int main() {
    double weight, height, bmi;

    printf("Enter weight in kilograms: ");
    if (scanf("%lf", &weight) != 1) {
        printf("Invalid input for weight.\n");
        return 1;
    }

    printf("Enter height in meters (e.g. 1.75): ");
    if (scanf("%lf", &height) != 1 || height <= 0) {
        printf("Invalid input for height.\n");
        return 1;
```



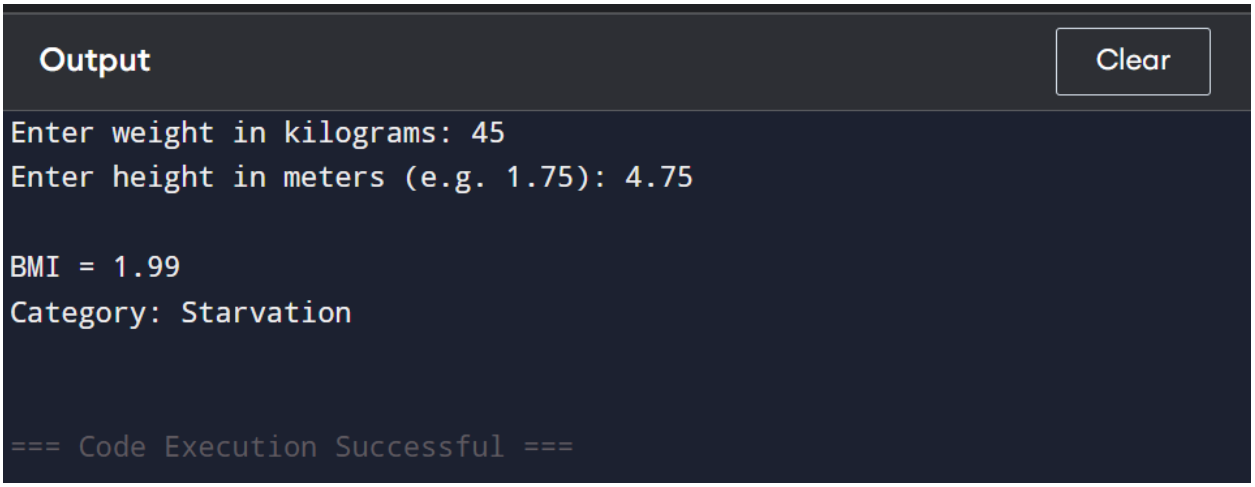
```c
    }

    bmi = weight / (height * height);

    printf("\nBMI = %.2f\n", bmi);

    /* Categories (continuous ranges so every bmi hits one category):
        Starvation      : < 15.0
        Anorexic        : 15.0 - 17.5
        Underweight     : 17.6 - 18.5
        Ideal           : 18.6 - 24.9
        Overweight      : 25.0 - 29.9
        Obese           : 30.0 - 39.9
```



```c
        Morbidity Obese : >= 40.0
    */

    if (bmi < 15.0) {
        printf("Category: Starvation\n");
    } else if (bmi >= 15.0 && bmi <= 17.5) {
        printf("Category: Anorexic\n");
    } else if (bmi > 17.5 && bmi <= 18.5) {
        printf("Category: Underweight\n");
    } else if (bmi > 18.5 && bmi <= 24.9) {
        printf("Category: Ideal\n");
    } else if (bmi >= 25.0 && bmi <= 29.9) {
        printf("Category: Overweight\n");
    } else if (bmi >= 30.0 && bmi <= 39.9) {
        printf("Category: Obese\n");
```



```c
    } else { /* bmi >= 40.0 */
        printf("Category: Morbidity Obese\n");
    }

    return 0;
}

```



```
Enter weight in kilograms: 45
Enter height in meters (e.g. 1.75): 4.75

BMI = 1.99
Category: Starvation



=== Code Execution Successful ===
```

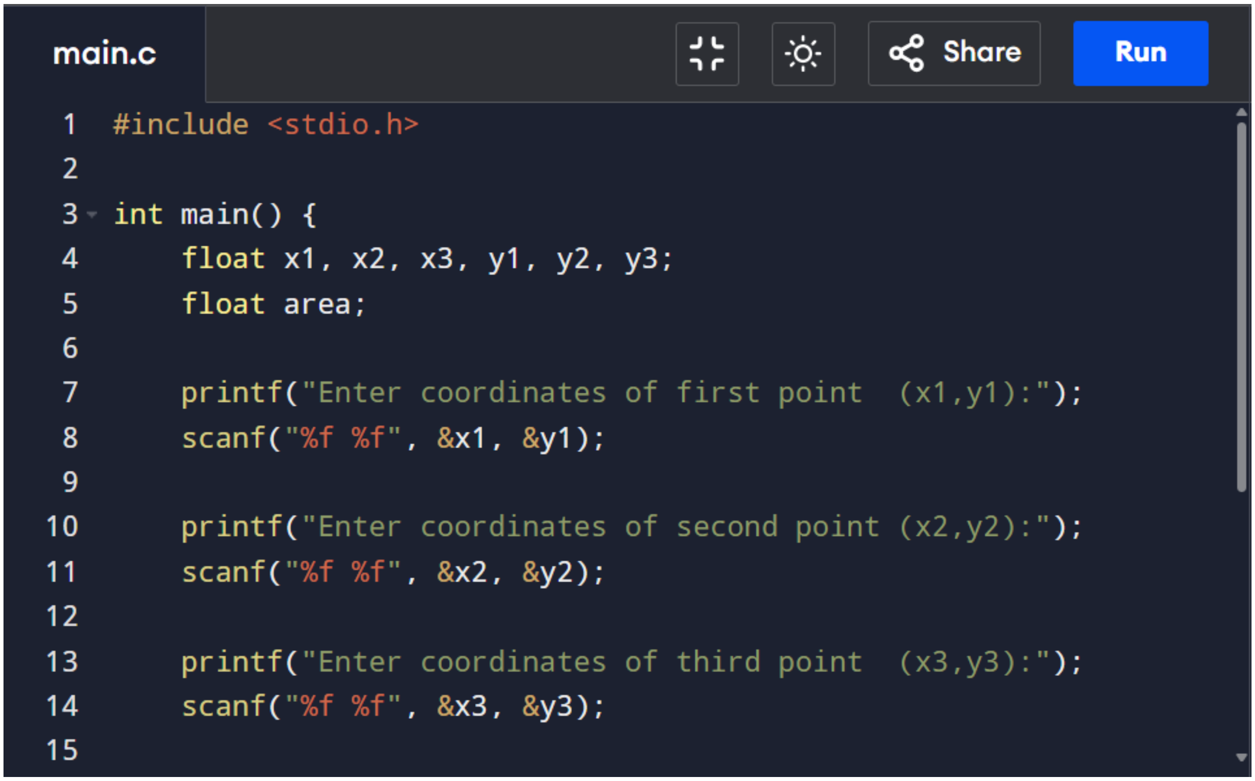
3.3. WAP to check if three points (x1,y1),(x2,y2) and (x3,y3) are collinear or not

```
1. Start
2. Input the coordinates $x1, y1, x2, y2, x3, y3$.
3. Calculate the area of the triangle formed
   by the three points using the formula:
       $\text{Area} = \frac{1}{2}[x1(y2 - y3) + x2(y3 - y1) + x3(y1 - y2)]$
4. If Area = 0, then the points are collinear.
5. Otherwise, the points are not collinear.
```

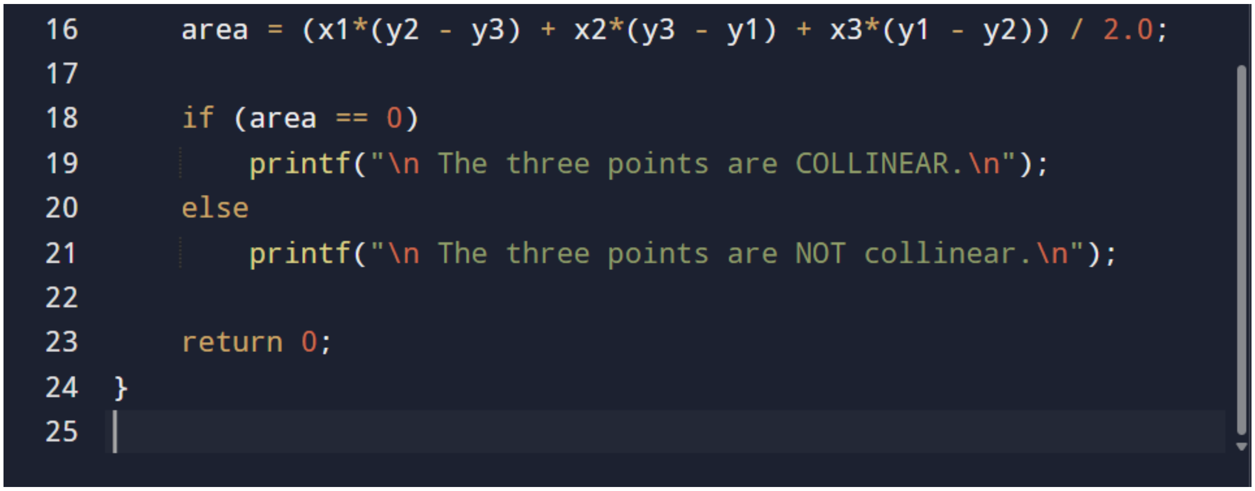
6. End

pseudo code

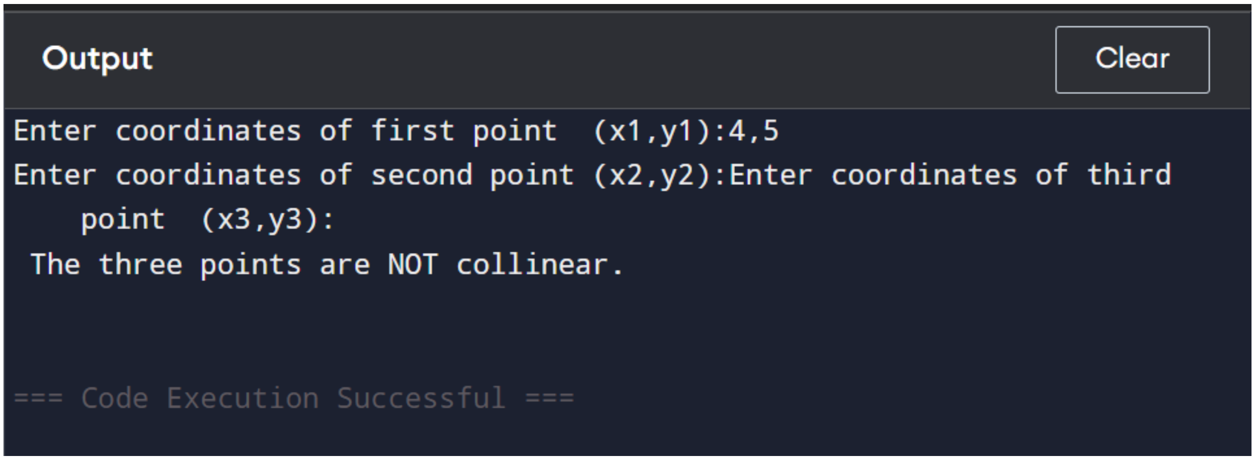
```
Start
   Input x1, y1
   Input x2, y2
   Input x3, y3

   area = 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
```



```c
#include <stdio.h>

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

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

    printf("Enter coordinates of second point (x2,y2):");
    scanf("%f %f", &x2, &y2);

    printf("Enter coordinates of third point  (x3,y3):");
    scanf("%f %f", &x3, &y3);

```



```c
    area = (x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2)) / 2.0;

    if (area == 0)
        printf("\n The three points are COLLINEAR.\n");
    else
        printf("\n The three points are NOT collinear.\n");

    return 0;
}

```



```
Enter coordinates of first point  (x1,y1):4,5
Enter coordinates of second point (x2,y2):Enter coordinates of third
    point  (x3,y3):
 The three points are NOT collinear.


=== Code Execution Successful ===
```

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 algorithim and pseudo code

Algorithim

□ Start

□ Input the year Y

□ **Initialize total_days = 0**

□ **Loop through all years from 1 to Y-1:**

- **If the year is a leap year (divisible by 4, not divisible by 100 unless divisible by 400), add 366 to total_days**
- **Else add 365 to total_days**

□ **Calculate day_index = (total_days + 1) % 7**

- **(We add 1 because 01/01/0001 is Monday → index 1)**

□ **Map day_index to day of week:**

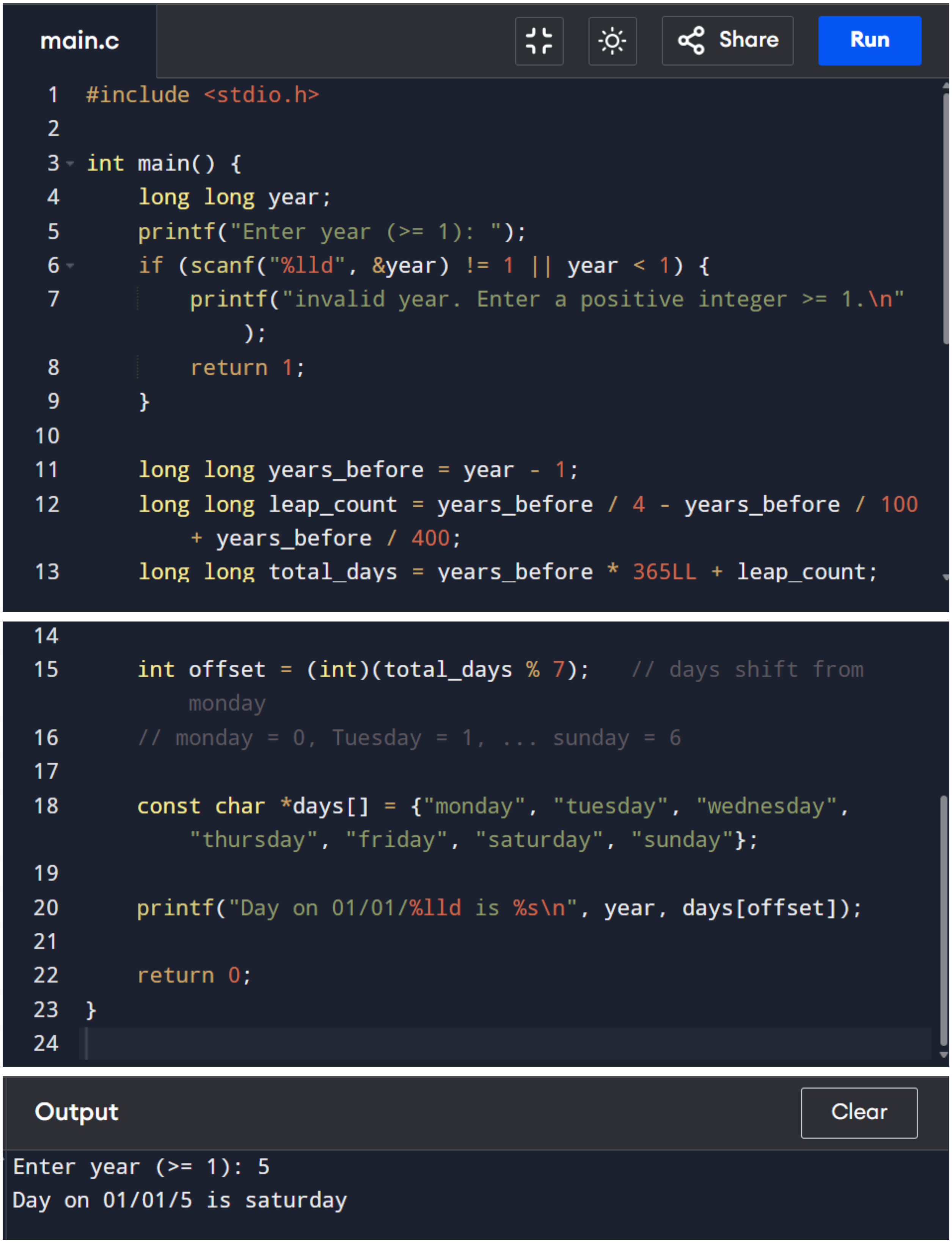
- **0 → Sunday, 1 → Monday, 2 → Tuesday, …, 6 → Saturday**

□ **Print the day of the week for 1st January of year Y**

□ **End**

```
START
  INPUT year Y
  total_days = 0
  FOR i = 1 TO Y-1
      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

  day_index = (total_days + 1) % 7  // 01/01/0001 was Monday (index 1)

  SWITCH(day_index)
      CASE 0: PRINT "Sunday"
      CASE 1: PRINT "Monday"
      CASE 2: PRINT "Tuesday"
      CASE 3: PRINT "Wednesday"
      CASE 4: PRINT "Thursday"
      CASE 5: PRINT "Friday"
      CASE 6: PRINT "Saturday"
  END SWITCH
```

END



```c
#include <stdio.h>

int main() {
    long long year;
    printf("Enter year (>= 1): ");
    if (scanf("%lld", &year) != 1 || year < 1) {
        printf("invalid year. Enter a positive integer >= 1.\n");
        return 1;
    }

    long long years_before = year - 1;
    long long leap_count = years_before / 4 - years_before / 100 + years_before / 400;
    long long total_days = years_before * 365LL + leap_count;

    int offset = (int)(total_days % 7);   // days shift from monday
    // monday = 0, Tuesday = 1, ... sunday = 6

    const char *days[] = {"monday", "tuesday", "wednesday", "thursday", "friday", "saturday", "sunday"};

    printf("Day on 01/01/%lld is %s\n", year, days[offset]);

    return 0;
}

```

```
Enter year (>= 1): 5
Day on 01/01/5 is saturday
```

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 should be three

   Algorithm

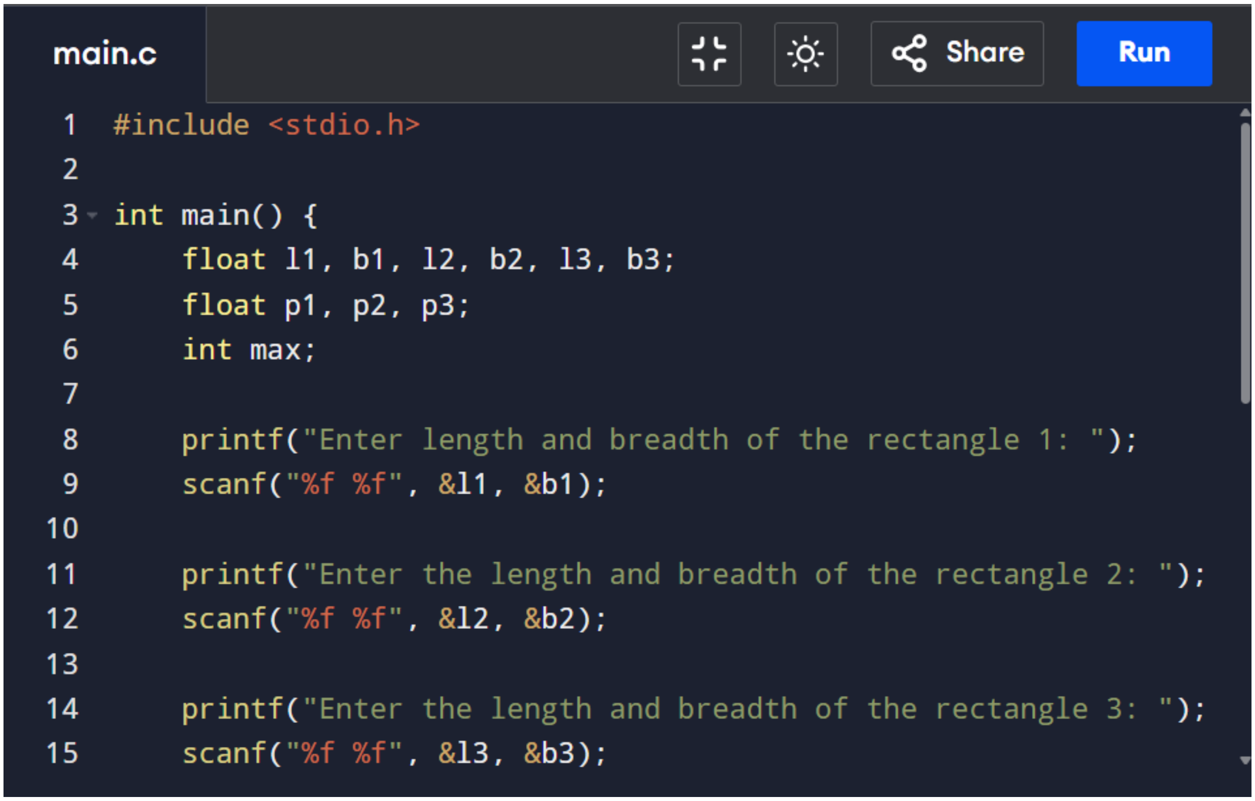
1. Start
2. Input length and breadth of Rectangle 1 → l1, b1
3. Input length and breadth of Rectangle 2 → l2, b2
4. Input length and breadth of Rectangle 3 → l3, b3
5. Calculate the perimeter of each rectangle:
   - p1 = 2 * (l1 + b1)
   - p2 = 2 * (l2 + b2)
   - p3 = 2 * (l3 + b3)
6. Use ternary operator to find the maximum perimeter:

  - max = (p1 > p2) ? ((p1 > p3) ? p1 : p3) : ((p2 > p3) ? p2 : p3)

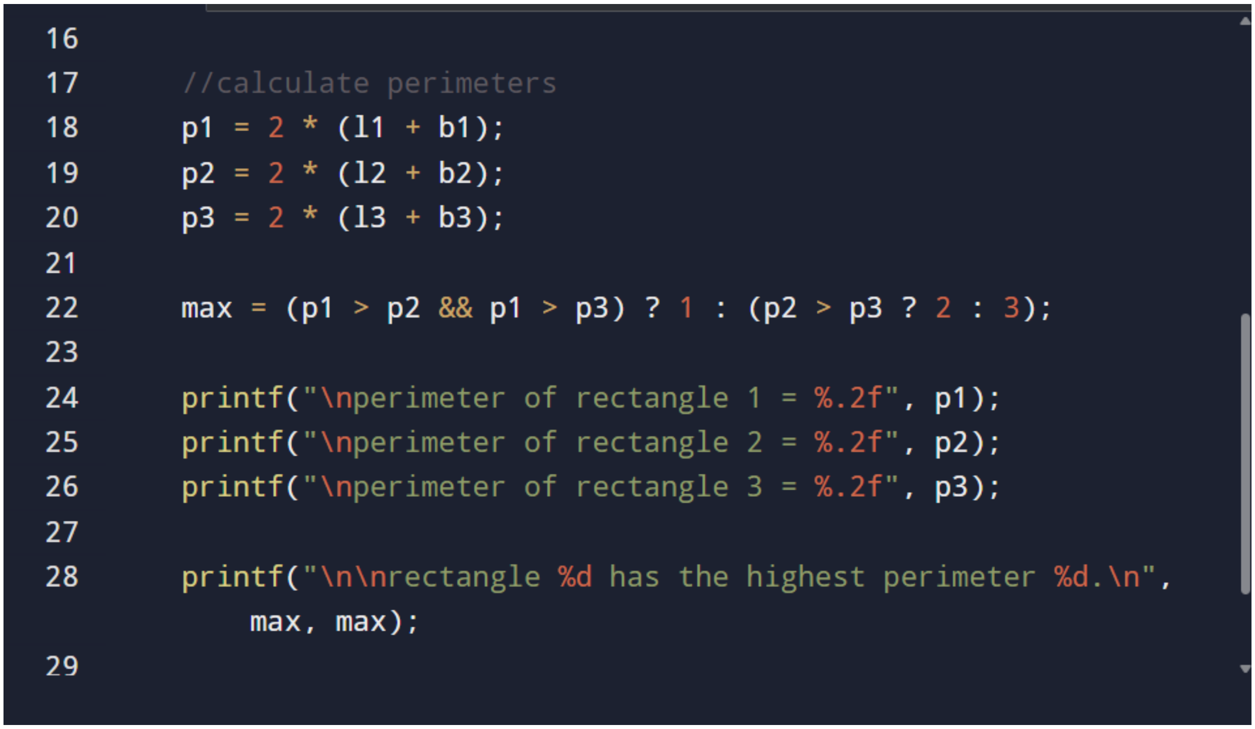
7. Print the rectangle with the highest perimeter and its value
8. End

Pseudo code

```
START
  // Input lengths and breadths of three rectangles
   INPUT l1, b1
   INPUT l2, b2
   INPUT l3, b3

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

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

  // Print result
  PRINT "Maximum perimeter rectangle =
", max
END
```

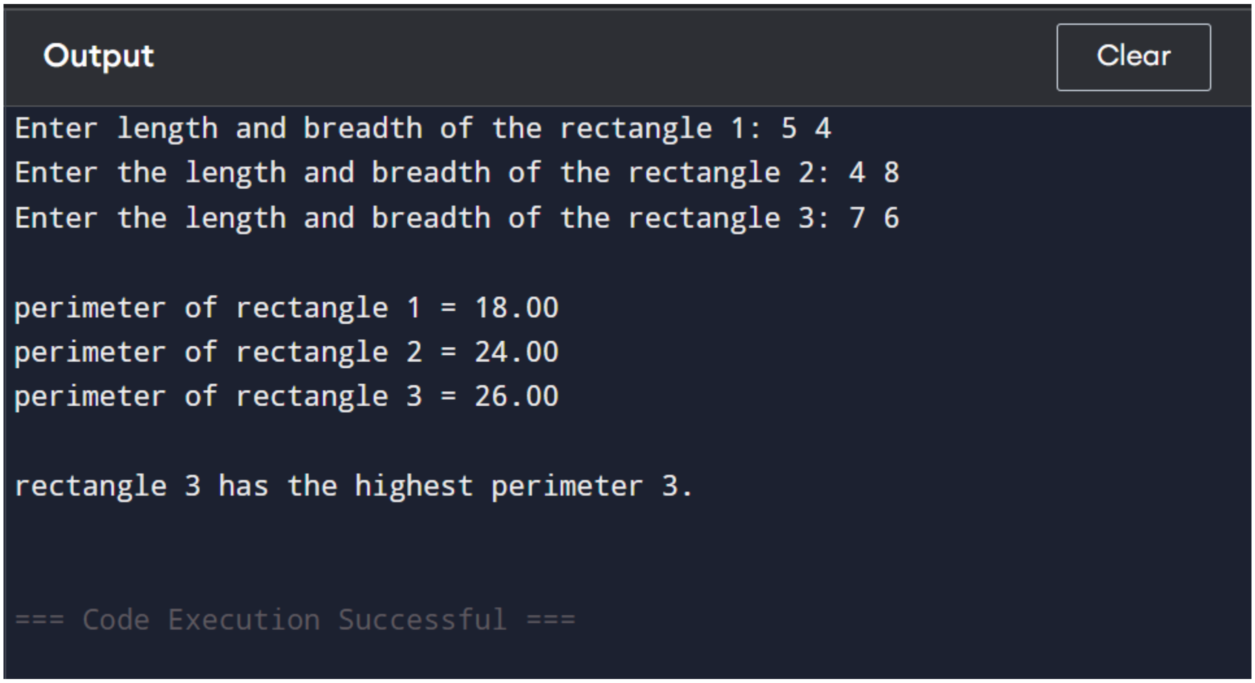


```c
#include <stdio.h>

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

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

    printf("Enter the length and breadth of the rectangle 2: ");
    scanf("%f %f", &l2, &b2);

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



```c

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

    max = (p1 > p2 && p1 > p3) ? 1 : (p2 > p3 ? 2 : 3);

    printf("\nperimeter of rectangle 1 = %.2f", p1);
    printf("\nperimeter of rectangle 2 = %.2f", p2);
    printf("\nperimeter of rectangle 3 = %.2f", p3);

    printf("\n\nrectangle %d has the highest perimeter %d.\n",
        max, max);

```

```

    return 0;
}

```



```
Enter length and breadth of the rectangle 1: 5 4
Enter the length and breadth of the rectangle 2: 4 8
Enter the length and breadth of the rectangle 3: 7 6

perimeter of rectangle 1 = 18.00
perimeter of rectangle 2 = 24.00
perimeter of rectangle 3 = 26.00

rectangle 3 has the highest perimeter 3.


=== Code Execution Successful ===
```

Experiment3.2:Loops

1. **WAP to enter number still the user wants. At the end, it should display the count of positive,negative,andZero esentered.**

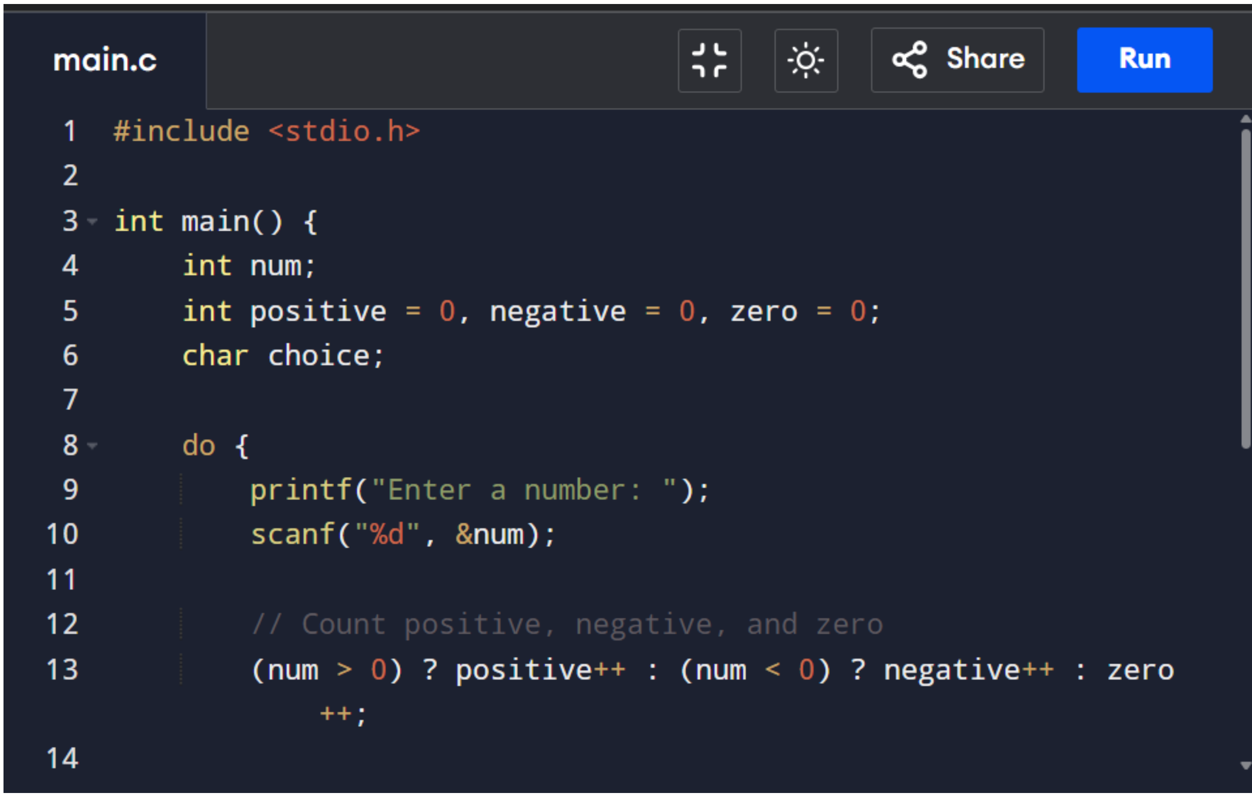
   Algorithm

1. **Start**

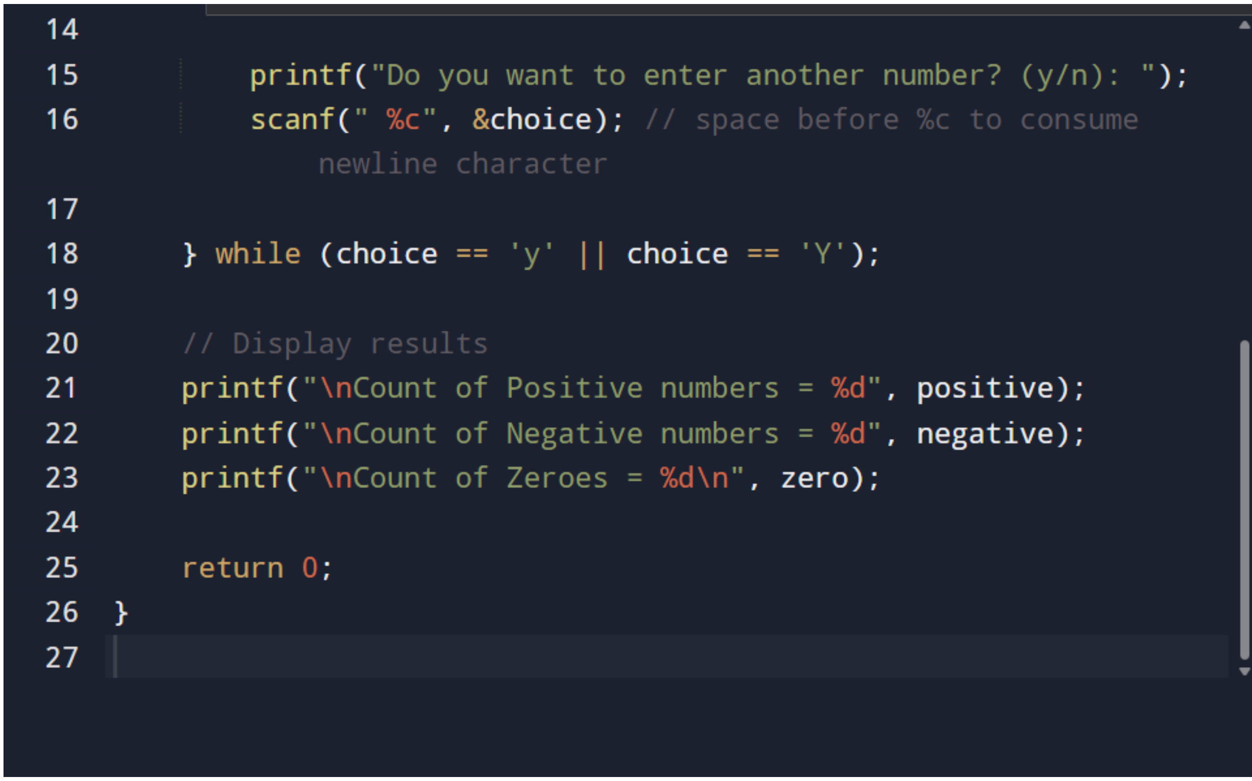
2. **Initialize counters: pos = 0, neg = 0, zero = 0**
3. **Repeat the following steps until the user wants to stop:**
    - **Input a number num**
    - **If num > 0 → increment pos**
    - **Else if num < 0 → increment neg**
    - **Else → increment zero**
    - **Ask the user if they want to continue (Yes/No)**
4. **After the loop ends, display the counts of positive, negative, and zero numbers**
5. **End**

Pseudo code

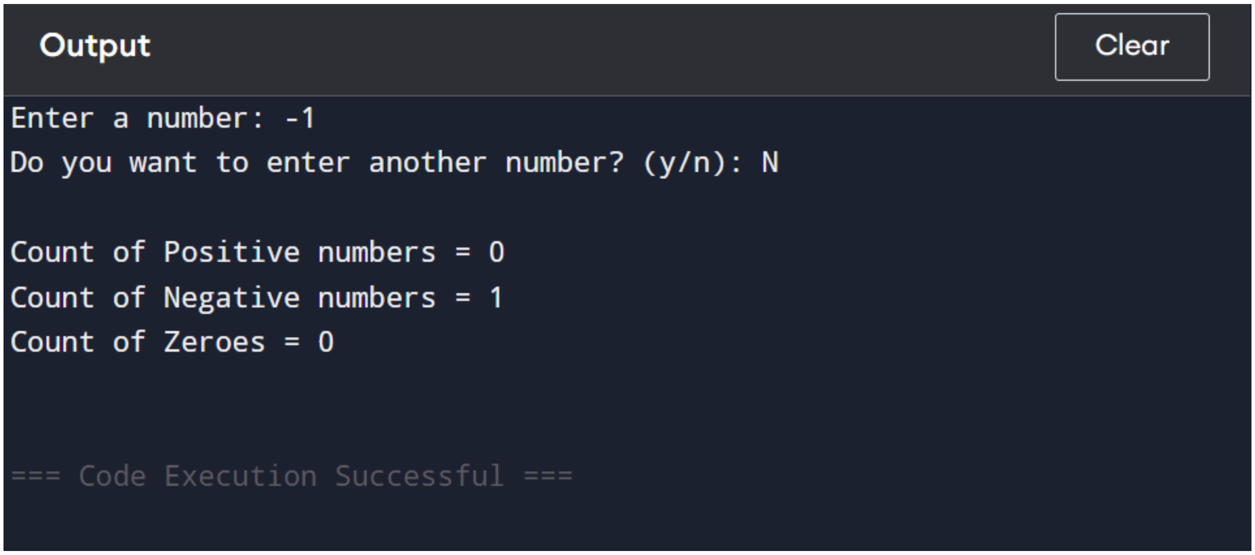
```
START
  pos = 0
  neg = 0
  zero = 0

  DO
    INPUT num
    IF num > 0 THEN
      pos = pos + 1
    ELSE IF num < 0 THEN
      neg = neg + 1
    ELSE
      zero = zero + 1
    END IF
    ASK user "Do you want to enter another number? (y/n)"
  WHILE user input is 'y'

  PRINT "Count of Positive numbers: ", pos
  PRINT "Count of Negative numbers: ", neg
  PRINT "Count of Zeroes: ", zero
END
```



```c
#include <stdio.h>

int main() {
    int num;
    int positive = 0, negative = 0, zero = 0;
    char choice;

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

        // Count positive, negative, and zero
        (num > 0) ? positive++ : (num < 0) ? negative++ : zero++;

```



```c

        printf("Do you want to enter another number? (y/n): ");
        scanf(" %c", &choice); // space before %c to consume newline character

    } while (choice == 'y' || choice == 'Y');

    // Display results
    printf("\nCount of Positive numbers = %d", positive);
    printf("\nCount of Negative numbers = %d", negative);
    printf("\nCount of Zeroes = %d\n", zero);

    return 0;
}

```



```
Enter a number: -1
Do you want to enter another number? (y/n): N

Count of Positive numbers = 0
Count of Negative numbers = 1
Count of Zeroes = 0


=== Code Execution Successful ===
```

2.Write a program to input a number from the user and print its multiplication table from 1 to 10 in the format:

Algorithim

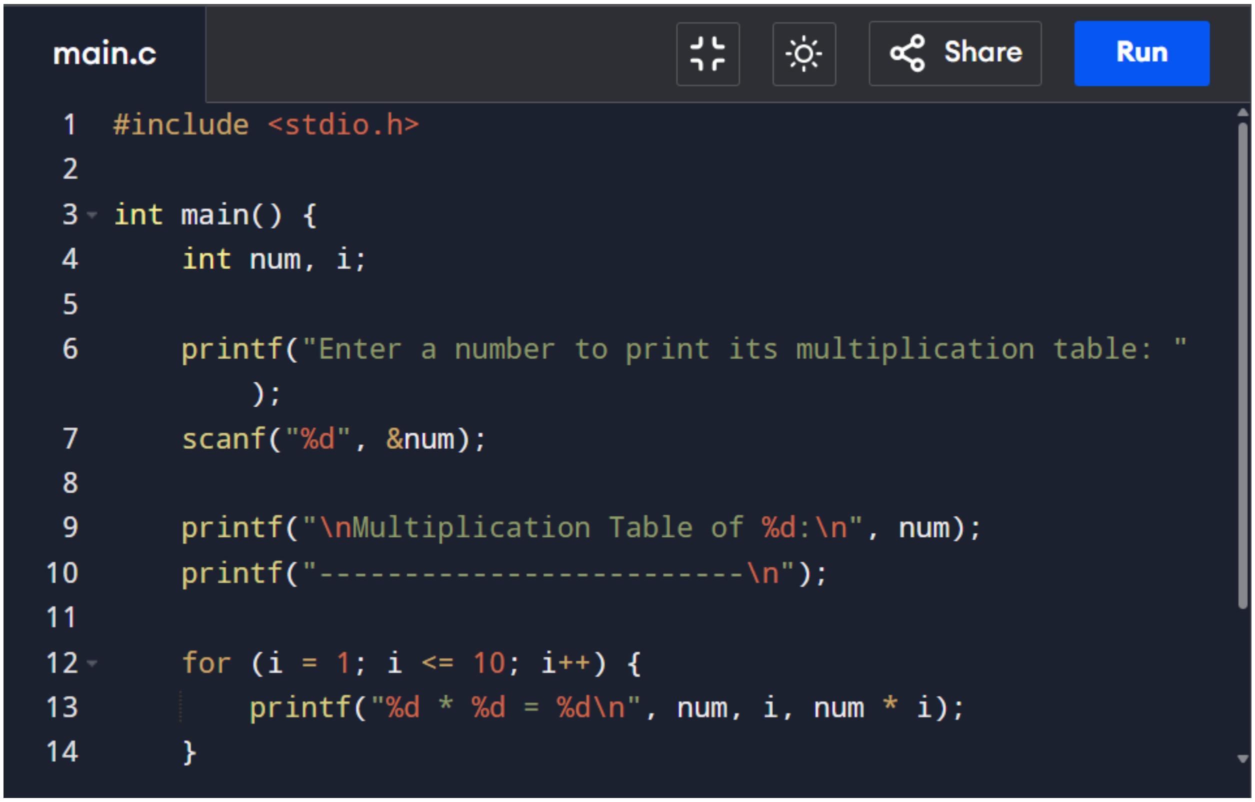
```
1. Start
2. Input a number n from the user
3. Initialize counter i = 1
4. Repeat the following steps while i <= 10:
    ○ Calculate result = n * i
    ○ Print n * i = result
```

   - **Increment i by 1**

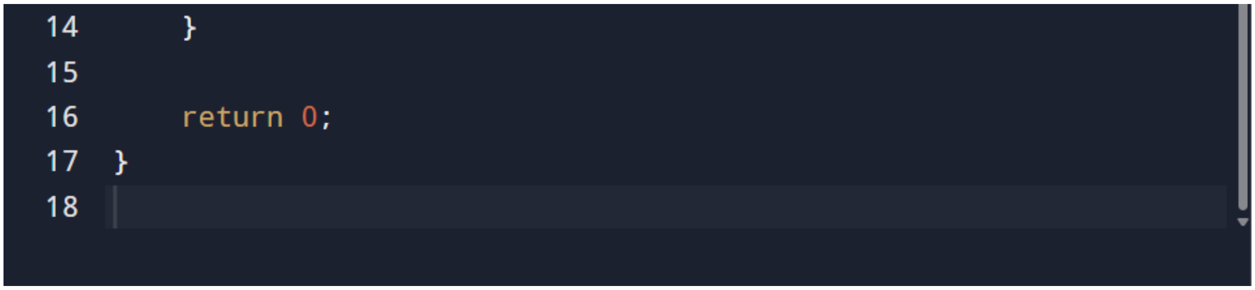
5. **End**

Pseudo code

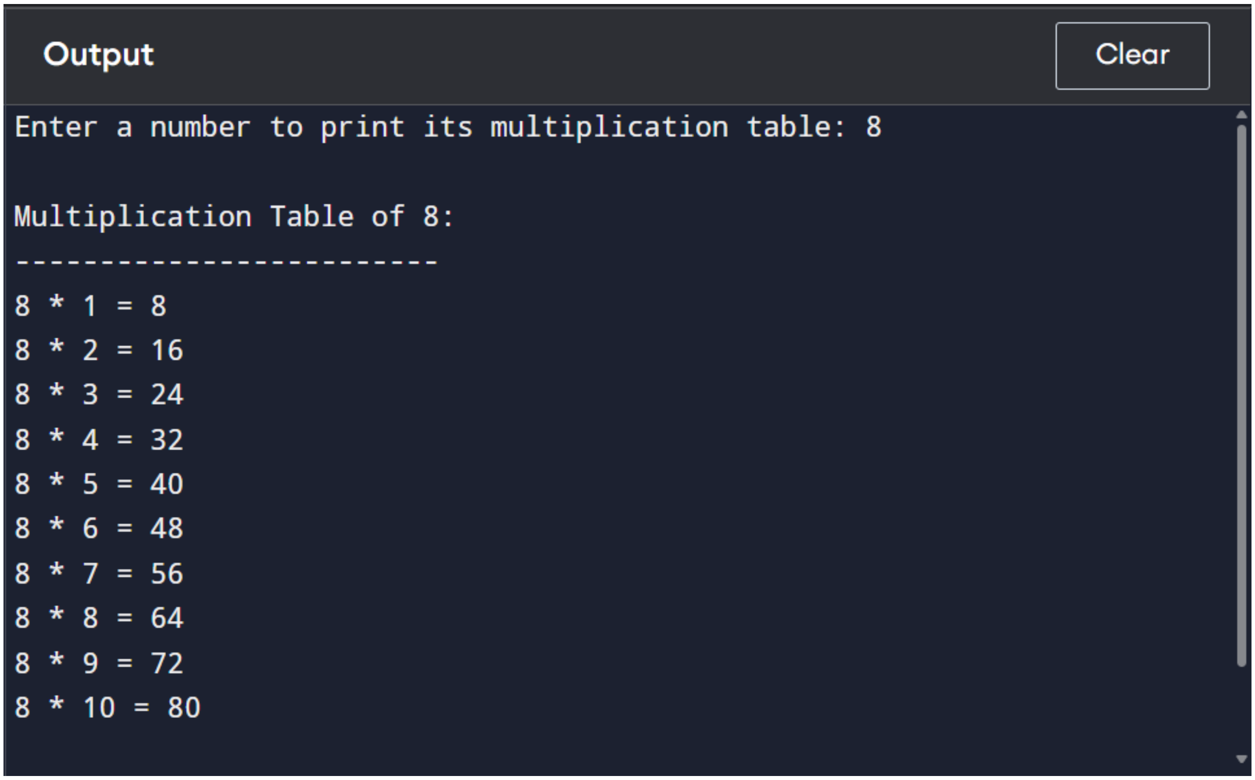
```
START
  INPUT n
  i = 1
  WHILE i <= 10
      result = n * i
      PRINT n, "*", i, "=", result
      i = i + 1
  END WHILE
END
```



```c
#include <stdio.h>

int main() {
    int num, i;

    printf("Enter a number to print its multiplication table: ");
    scanf("%d", &num);

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

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



```c
    }

    return 0;
}
```



```
Enter a number to print its multiplication table: 8

Multiplication Table of 8:
-------------------------
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
```