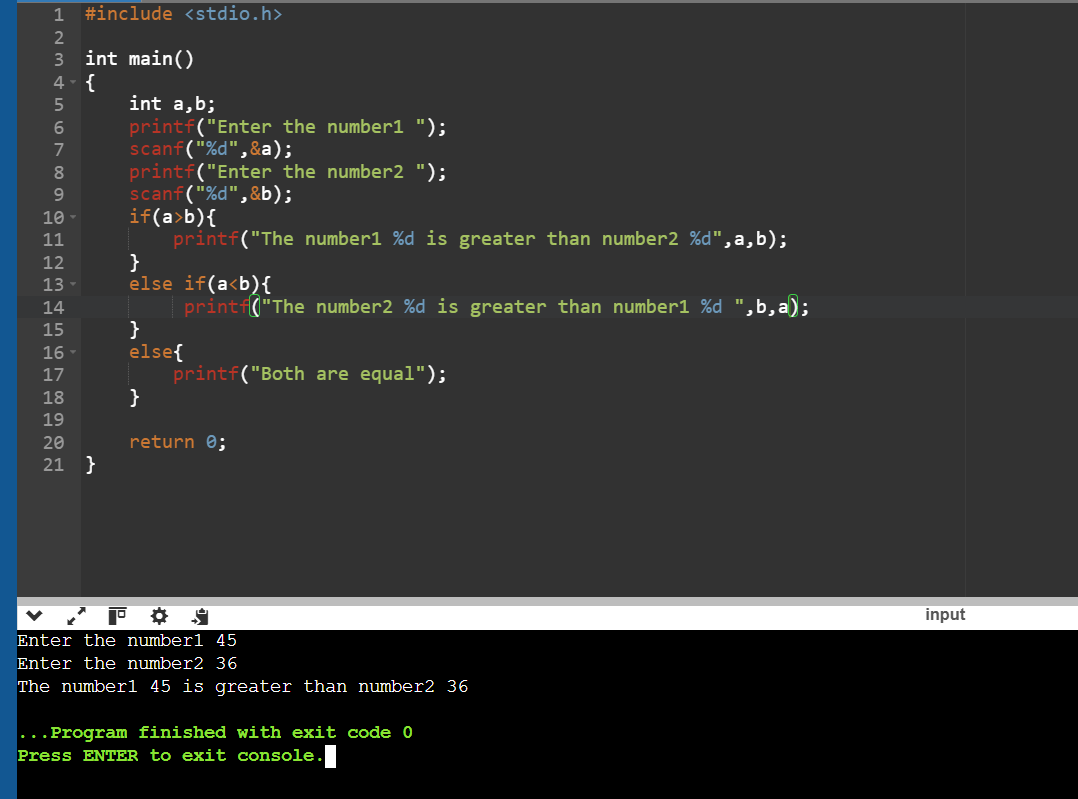
1. **Equality Check:**  
   Write a program to check if two integers provided by the user are equal or not.

A screenshot of a computer program

Description automatically generated

1. **Greater Number Identification:**  
   Write a program to determine which of two numbers is greater using relational operators.

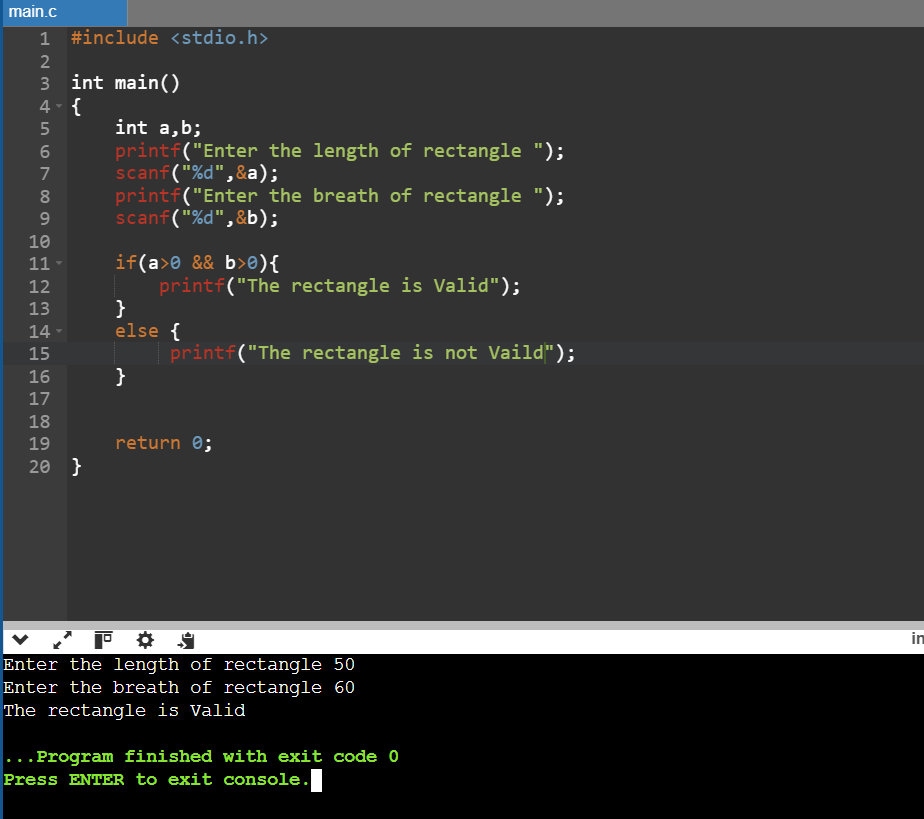


1. **Check if a Number is Positive:**  
   Use relational operators to check if a given number is positive (greater than 0).

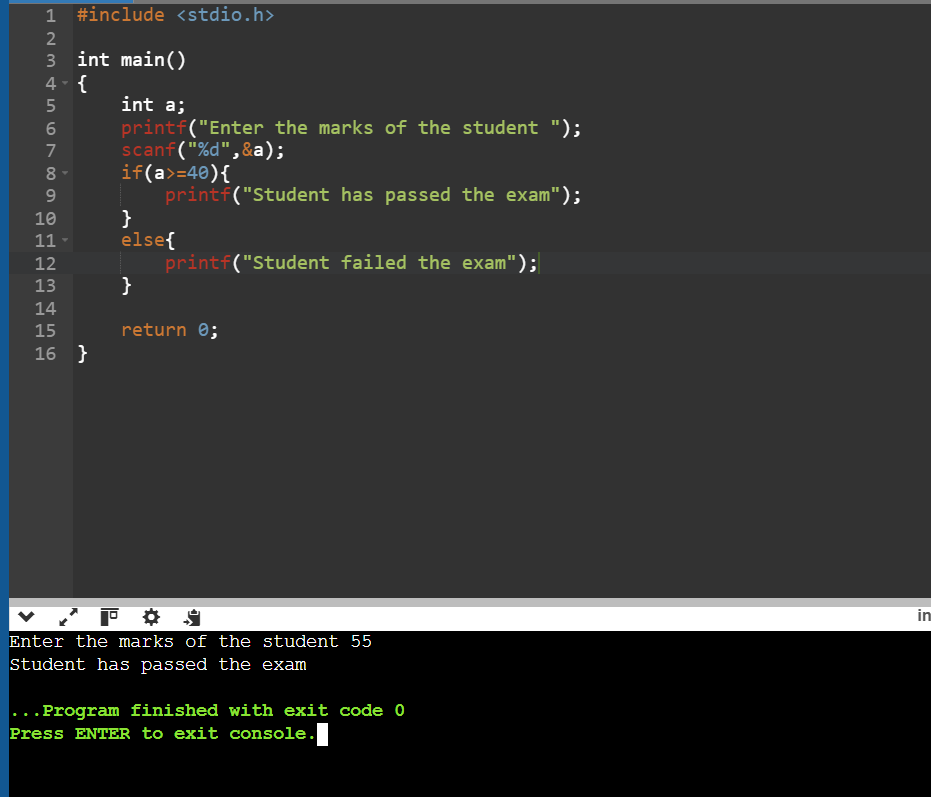
A screen shot of a computer program

Description automatically generated

1. **Rectangle Validity Check:**  
   Write a program to verify if the given length and breadth of a rectangle satisfy the condition of a valid rectangle (length > 0 and breadth > 0).



1. **Grade Eligibility Check:**  
   Given a student's marks in a subject, determine if the student has passed (marks >= 40).

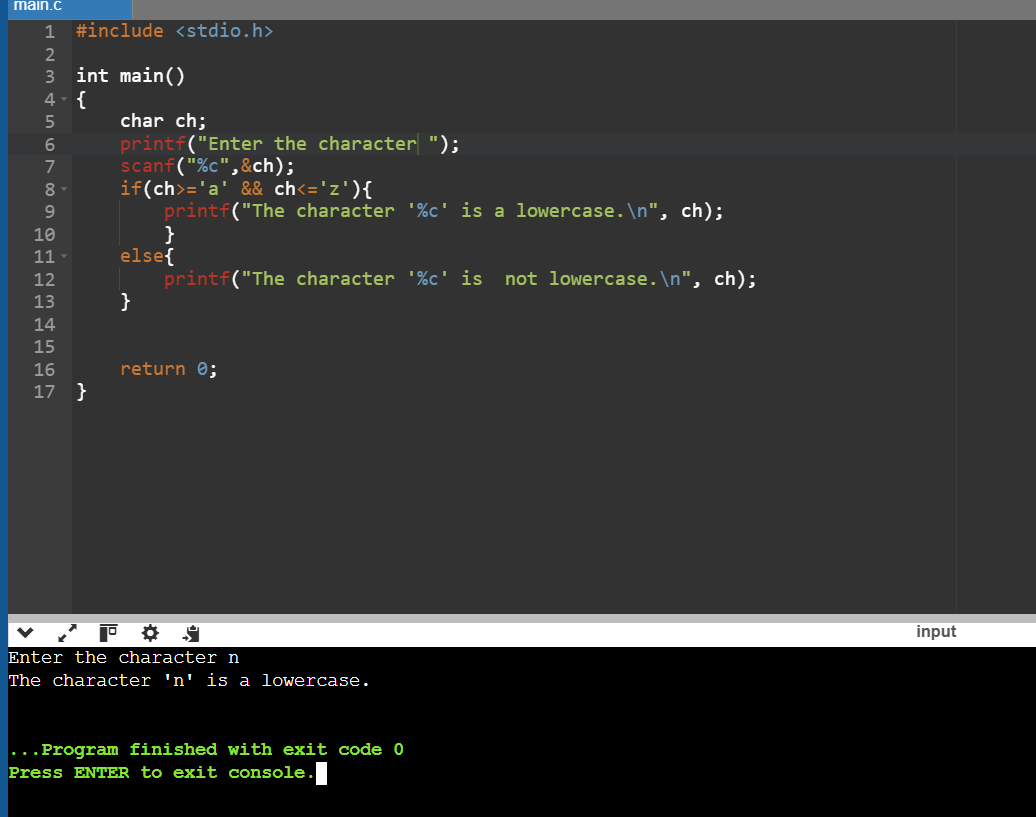


1. **Check if Number is Within Range:**  
   Use relational operators to check if a given number lies between 10 and 50 (inclusive).

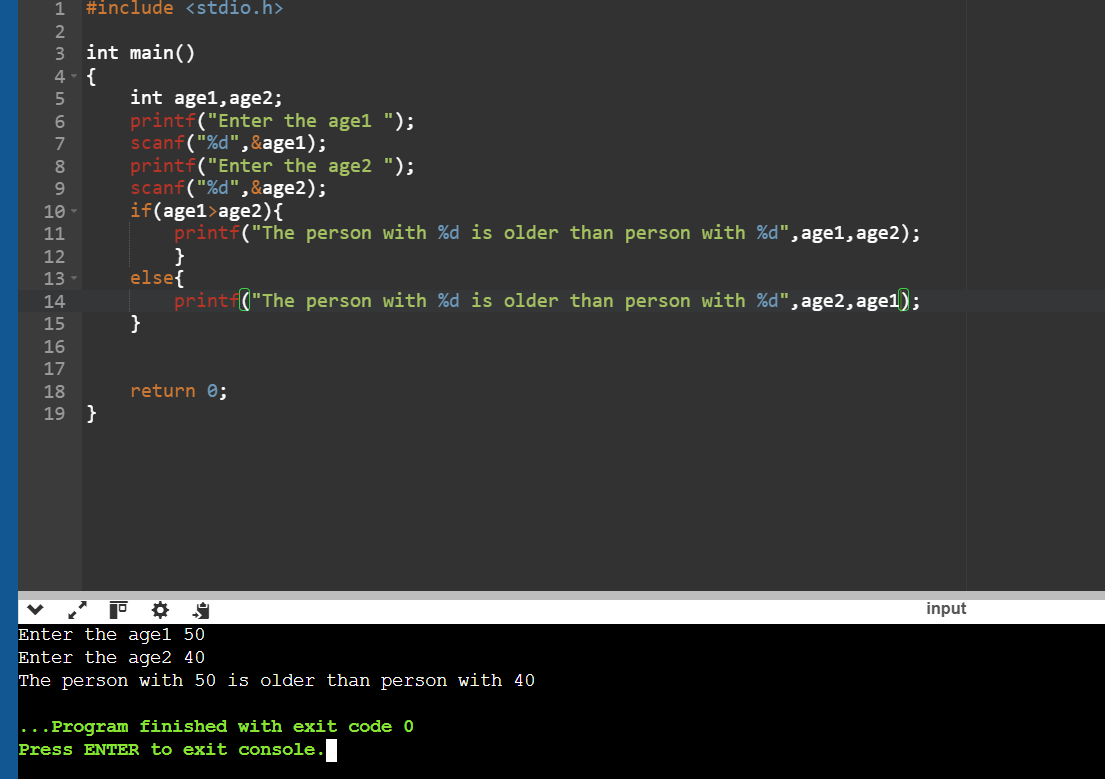
A screenshot of a computer program

Description automatically generated

1. **Verify Alphabetic Range:**  
   Write a program to check if a given character is a lowercase English letter (between 'a' and 'z').



1. **Age Comparison:**  
   Compare the ages of two people and determine who is older or if both are of the same age.

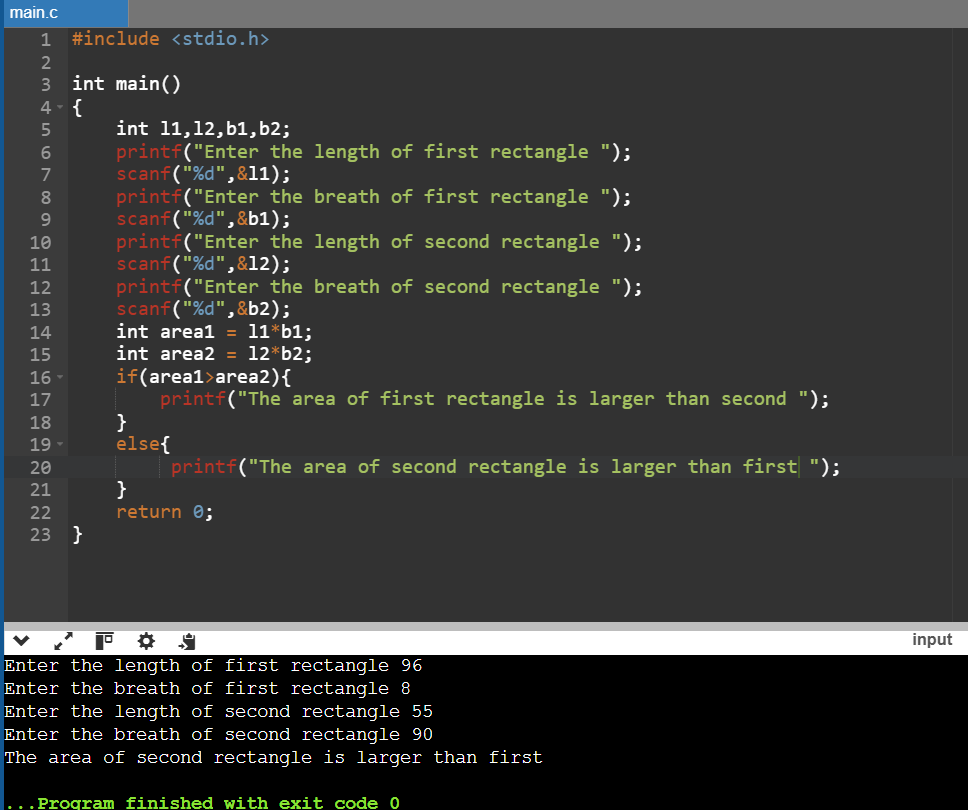


1. **Weight Limit Check:**  
   Write a program to determine if the weight of an object exceeds the specified maximum limit (e.g., 50 kg).

A screenshot of a computer program

Description automatically generated

1. **Rectangle Larger Area Check:**  
   Compare the areas of two rectangles given their lengths and breadths and determine which rectangle has a larger area.



1. Write a program to compute the result of the bitwise AND operation between two integers provided by the user.

A screen shot of a computer program

Description automatically generated

1. Write a program to compute the result of the bitwise OR operation between two integers provided by the user.

A screenshot of a computer program

Description automatically generated

1. Write a program to compute the result of the bitwise XOR operation between two integers provided by the user.

A screen shot of a computer program

Description automatically generated

1. Write a program to find the bitwise complement of a given integer and print the result.

A screenshot of a computer program

Description automatically generated

1. Given an integer n and a position p, write a program to toggle the bit at position p using the XOR operator.

A screen shot of a computer program

Description automatically generated

1. Write a program to set the bit at a given position p in an integer n to 1 using the OR operator.

A screenshot of a computer program

Description automatically generated

1. Write a program to clear (set to 0) the bit at a given position p in an integer n using the AND and NOT operators.

A screen shot of a computer program

Description automatically generated

1. **Number Properties Validation:**  
   Write a program to check if a given integer is both a multiple of 5 (arithmetic operator) and greater than 50 (relational operator). Additionally, verify if its binary representation has its least significant bit set (bitwise AND operation).

A screen shot of a computer

Description automatically generated

1. **Toggle and Evaluate Bit Status:**  
   Given an integer n and a bit position p:
   * Use bit masking and bitwise XOR to toggle the bit at position p.
   * After toggling, check if the updated number is positive (arithmetic and relational operators) and divisible by 2 (logical operators).

A screenshot of a computer program

Description automatically generated

1. **Determine Voting Eligibility with Criteria:**  
   A person can vote if:
   * Their age is greater than or equal to 18 (relational operator).
   * They are a registered citizen, represented by a specific bit set in their ID number (bit masking and bitwise AND).  
     Write a program to verify these conditions using logical operators.

A screen shot of a computer program

Description automatically generated

1. **Set, Clear, and Check Specific Bit:**  
   Write a program to:
   * Use bit masking and bitwise OR to set a specific bit in a number.
   * Use bitwise AND and NOT to clear another specific bit.
   * Check if the resulting number is odd (arithmetic and relational operators) and lies within a range (logical operators).

A screen shot of a computer program

Description automatically generated

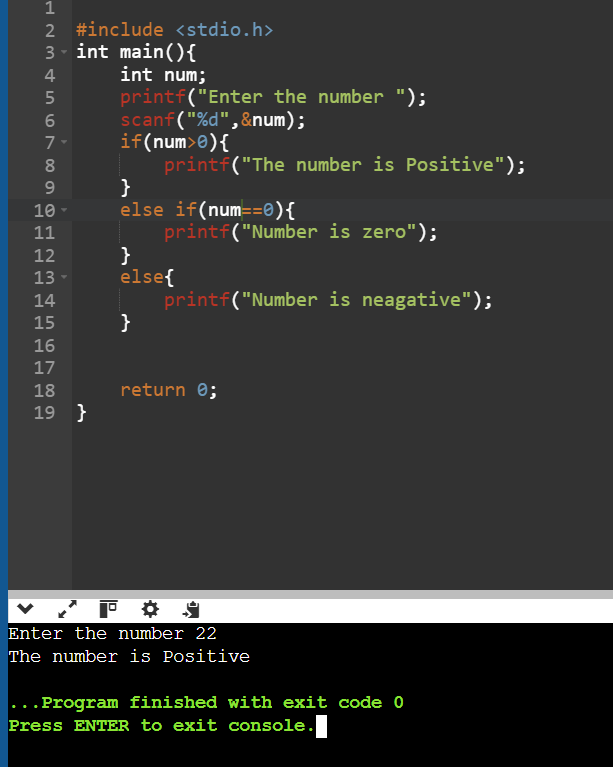
1. **Custom Mathematical Condition with Bits:**  
   Given two integers a and b, perform the following:
   * Compute their sum and product (arithmetic operators).
   * Verify if the sum is greater than 100 and the product is divisible by 4 (relational and logical operators).
   * Check if the binary representation of a has its second bit set (bitwise AND with a mask).

A screenshot of a computer program

Description automatically generated

**If Statements**

1. **Check for Positivity:**  
   Write a program to check if a number entered by the user is positive using an **if** statement.



1. **Divisibility Check:**  
   Write a program to check if a number is divisible by 3 using an **if** statement.

A screen shot of a computer program

Description automatically generated

**If-Else Statements**

1. **Odd or Even:**  
   Write a program to determine if a number is odd or even using an **if-else** statement.

A screen shot of a computer program

Description automatically generated

1. **Passing Criteria:**  
   Write a program to check if a student has passed an exam based on their marks (pass marks are 40). If the marks are below 40, display "Fail."

A screen shot of a computer program

Description automatically generated

**Nested If-Else Statements**

1. **Triangle Type Checker:**  
   Given the lengths of three sides, write a program to determine if the triangle is valid using nested **if-else**. If valid, check if it is an equilateral triangle.

A screen shot of a computer program

Description automatically generated

1. **Eligibility for Admission:**  
   Write a program to check if a student is eligible for admission based on the following criteria:
   * Marks in mathematics >= 50
   * Marks in physics >= 50
   * Total marks (math + physics) >= 120  
     Use nested **if-else** statements.

A screenshot of a computer program

Description automatically generated

**If-Else-If Ladder**

1. **Grade Calculator:**  
   Write a program to calculate and print the grade of a student based on their percentage using an **if-else-if ladder**:
   * = 90: Grade A
   * = 75: Grade B
   * = 50: Grade C
   * < 50: Fail

A screen shot of a computer program

Description automatically generated

1. **Number Classification:**  
   Write a program to classify an integer as positive, negative, or zero using an **if-else-if ladder**.

A screenshot of a computer program

Description automatically generated

1. **Electricity Bill Calculation:**  
   Write a program to calculate the electricity bill based on the number of units consumed using the following criteria:
   * Units <= 100: ₹5 per unit
   * Units > 100 and <= 200: ₹7 per unit
   * Units > 200: ₹10 per unit  
     Use an **if-else-if ladder** to implement this.

A screen shot of a computer program code

Description automatically generated

1. **Day of the Week:**  
   Write a program to print the name of the day of the week based on a number entered by the user (1 for Monday, 2 for Tuesday, ..., 7 for Sunday) using an **if-else-if ladder**.

A screen shot of a computer program

Description automatically generated

**Switch Case**

1. Write a program that takes an integer (1-7) as input and uses a switch-case to print the corresponding day of the week (e.g., 1 for Monday, 2 for Tuesday, etc.).

A screen shot of a computer program

Description automatically generated

1. Write a program to perform basic arithmetic operations (addition, subtraction, multiplication, division) based on the operator input (+, -, \*, /) using a switch-case statement.

A screenshot of a computer program

Description automatically generated

1. Write a program that takes a single character as input and uses a switch-case to determine if it is a vowel or a consonant.

A screen shot of a computer program

Description automatically generated

1. Write a program to convert a single-digit number (0-9) into its word representation (e.g., 1 to "One", 2 to "Two") using a switch-case statement.

A screen shot of a computer program

Description automatically generated

1. Write a program that takes an integer (1-12) as input and uses a switch-case to print the name of the corresponding month (e.g., 1 for January, 2 for February, etc.).

A screen shot of a computer program

Description automatically generated

1. Write a program that takes a grade (A, B, C, D, F) as input and uses a switch-case to print the description of the grade (e.g., A: "Excellent", B: "Good", etc.).

A screen shot of a computer program

Description automatically generated

1. Write a menu-driven program that offers the user options for basic mathematical operations (addition, subtraction, etc.). Based on the user’s choice, perform the corresponding operation using a switch-case.

A screen shot of a computer program

Description automatically generated

1. Write a program to simulate a traffic light system. Take input as R, Y, or G (Red, Yellow, Green) and use a switch-case to display the corresponding action (e.g., R for Stop, Y for Get Ready, G for Go).

A screen shot of a computer program

Description automatically generated

1. Write a program that takes the year as input and uses a switch-case to check and print whether it is a leap year or not (use logical division by 4 and additional logic in cases).

A screenshot of a computer program

Description automatically generated

1. Write a program to calculate the area of different shapes based on user input:
   * 1 for Circle
   * 2 for Rectangle
   * 3 for Triangle  
     Use a switch-case to perform the respective area calculations.

A computer screen shot of a program

Description automatically generated