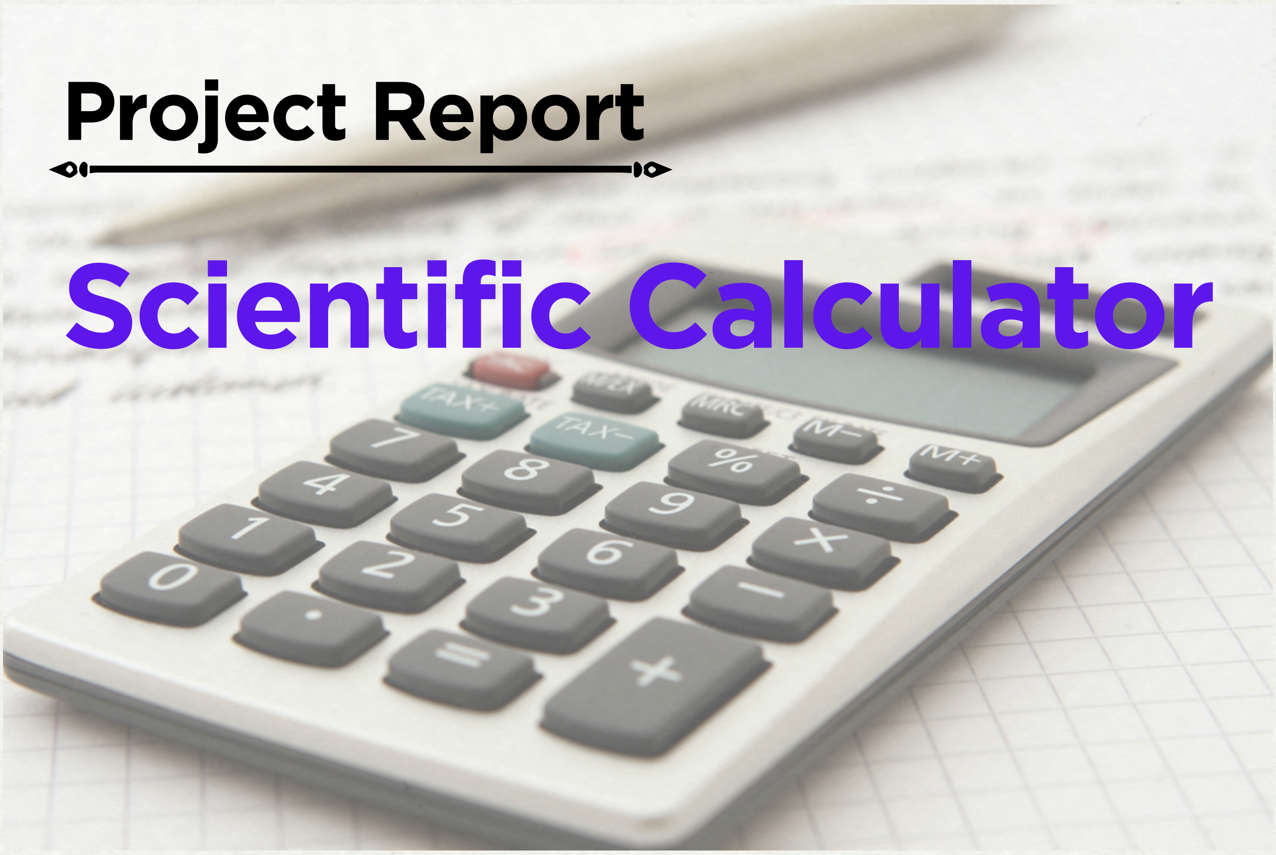
Project Report



Scientific Calculator

Name	**Rohit Raj**
Sec	**E**
Roll number	56
Stream	CSE[IOTCSBT]
Enrollment No.	12022002017043

ESC103(Pr.) **Programming for problem solving using C**

VARIABLE DESCRIPTION

- `choice` - an integer variable that stores the user's choice of operation (e.g., addition, subtraction, etc.) from the menu
- `num1` - a double variable that stores the first operand entered by the user for arithmetic operations
- `num2` - a double variable that stores the second operand entered by the user for arithmetic operations
- `result` - a double variable that stores the result of the arithmetic operation performed by the program
- `input` - a double variable that stores the number entered by the user for scientific calculations like square root, power, and logarithm
- `base` - a double variable that stores the base entered by the user for logarithm calculation
- `error_flag` - a boolean variable that stores the error status of the program, which is set to `true` if any error occurs during the operation and `false` otherwise

Functions Description

1. **printf() - This function is used to print output to the console.**
2. **scanf() - This function is used to read input from the user via the console.**
3. **pow() - This function calculates the result of raising a number to a given power.**
4. **sqrt() - This function calculates the square root of a given number.**
5. **sin() - This function calculates the sine of an angle in radians.**
6. **cos() - This function calculates the cosine of an angle in radians.**
7. **tan() - This function calculates the tangent of an angle in radians.**
8. **M_PI - This is a constant defined in the math.h header file, representing the value of pi.**

Files Description

The C program provided is a basic implementation of a scientific calculator. The program prompts the user to choose an operation to perform from a list of available options, and then takes input from the user to perform the selected operation.

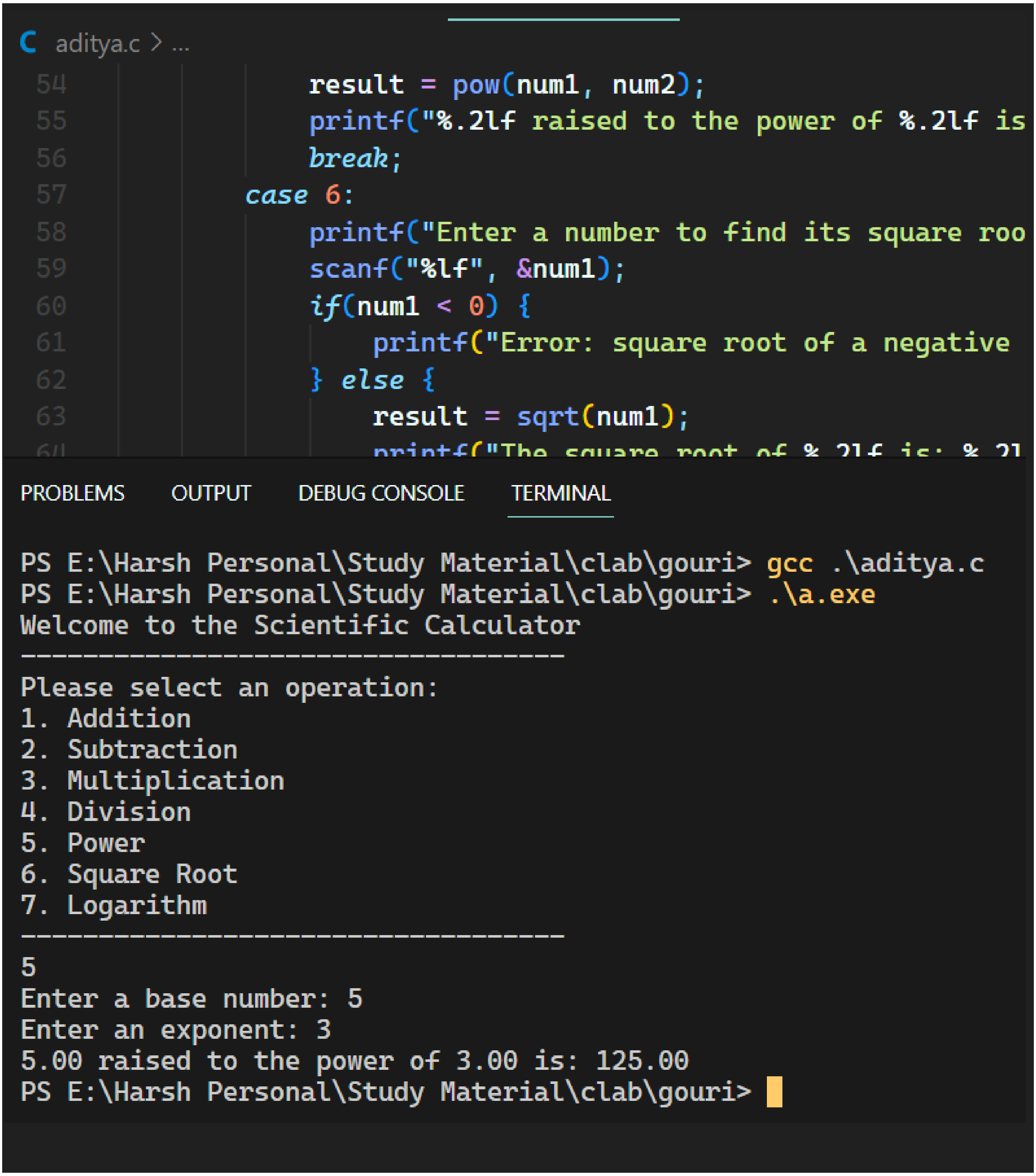
The available operations include basic arithmetic operations such as addition, subtraction, multiplication, and division of two numbers. Additionally, the program also allows for exponentiation and calculation of square roots. Finally, the program also offers the calculation of trigonometric functions such as sine, cosine, and tangent.

The program uses a switch statement to perform the selected operation based on the user's input. The user is prompted to input values for the operations that require input, such as entering two numbers for arithmetic operations and specifying the base and exponent for exponentiation.

The program also includes error handling for invalid inputs. For example, the program displays an error message when attempting to take the square root of a negative number or divide by zero.

Overall, this program provides a basic implementation of a scientific calculator with several common mathematical operations.

Output



```
                result = pow(num1, num2);
                printf("%.2lf raised to the power of %.2lf is
                break;
            case 6:
                printf("Enter a number to find its square roo
                scanf("%lf", &num1);
                if(num1 < 0) {
                    printf("Error: square root of a negative
                } else {
                    result = sqrt(num1);
                    printf("The square root of %.2lf is: %.2l
```

```
PS E:\Harsh Personal\Study Material\clab\gouri> gcc .\aditya.c
PS E:\Harsh Personal\Study Material\clab\gouri> .\a.exe
Welcome to the Scientific Calculator
-----------------------------------
Please select an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Power
6. Square Root
7. Logarithm
-----------------------------------
5
Enter a base number: 5
Enter an exponent: 3
5.00 raised to the power of 3.00 is: 125.00
PS E:\Harsh Personal\Study Material\clab\gouri>
```