ULAB

UNIVERSITY OF LIBERAL ARTS BANGLADESH

# Open Ended Lab 1

Fall 2024

**Course Title: Structured Programming Lab**

**Course Code: CSE 1202 (Fall 2024)**

**Submitted by: Student**

**Name and ID**

Md. Tazminur Rahman Tanim (242014124)

**Department of CSE**

## University of Liberal Arts Bangladesh (ULAB)

1. Define the requirements and features of the menu-driven calculator program, including supported operations and input validation rules.

Answer :

**Define the Requirements and Features of the menu-driven calculator program,**
**Supported Operations**:

1. Addition: Perform the sum of two or more numbers.
2. Subtraction: Calculate the difference between two numbers.
3. Multiplication: Compute the product of two or more numbers.
4. Division: Divide one number by another, ensuring no division by zero occurs.
5. Percentage Calculation: Find the percentage of a number relative to a total.
6. Exit Option: Allow the user to exit the program when they wish.

**Input Validation Rules:**

1. Accept only numerical inputs (integers or floating-point numbers).
2. Reject non-numerical inputs with a proper error message.
3. Prevent division by zero with a clear warning message.
4. Ensure no runtime errors occur during operations.

**Features**

**User-Friendly Interface**:

1. Provide a menu with numbered options for easy navigation.
2. Display results clearly after each operation.

**Continuous Operation**:

1. The program will run continuously in a loop until the user chooses to exit.

**Error Handling**:

1. Provide meaningful error messages for invalid inputs or operations.
2. Handle edge cases such as zero in division or percentage total.

**Modular Design**:

1. Each operation is implemented as a separate function, ensuring reusability and readability.

**Flexibility**:

1. Support operations for any number of inputs in addition and multiplication.
2. Allow dynamic input for subtraction, division, and percentage.

2. Develop algorithms for each functionality (addition, subtraction, multiplication, division, percentage calculation, and input validation).

Answer :

**Algorithm Development :**

**Main Menu Algorithm**

1. Display the menu with operation choices.
2. Accept user input for the selected operation.
3. Call the appropriate function based on the choice.
4. Repeat until the user selects "Exit."

**Addition Function**

1. Start.
2. Prompt for the number of values.
3. Input numbers and calculate their sum.
4. Display the result.
5. End.

**Subtraction Function**

1. Start.
2. Input two numbers.
3. Subtract the second number from the first.
4. Display the result.
5. End.

**Multiplication Function**

1. Start.
2. Prompt for the number of values.
3. Input numbers and calculate their product.
4. Display the result.
5. End.

**Division Function**

1. Start.
2. Input two numbers.
3. Check if the second number is zero:
    a) If yes, display an error.
    b) If no, perform the division.

4. Display the result.

5. End

### Percentage Calculation Function

```
1. Start

2. Input a number and the total.

3. Check if the total is zero:

    a) If yes, display an error.

    b) If no, calculate the percentage.

4. Display the result.

5. End
```

### Validation Algorithm

```
1. Start.
2. Prompt the user for input.
3. Check if the input is a valid number:
    • If valid, return the number.
    • If invalid, display an error message and prompt again.
4.  Repeat until valid input is provided.
5.  End.
```

3. Implement the designed algorithms in a programming language of choice to create a fully functional calculator program.
Answer :

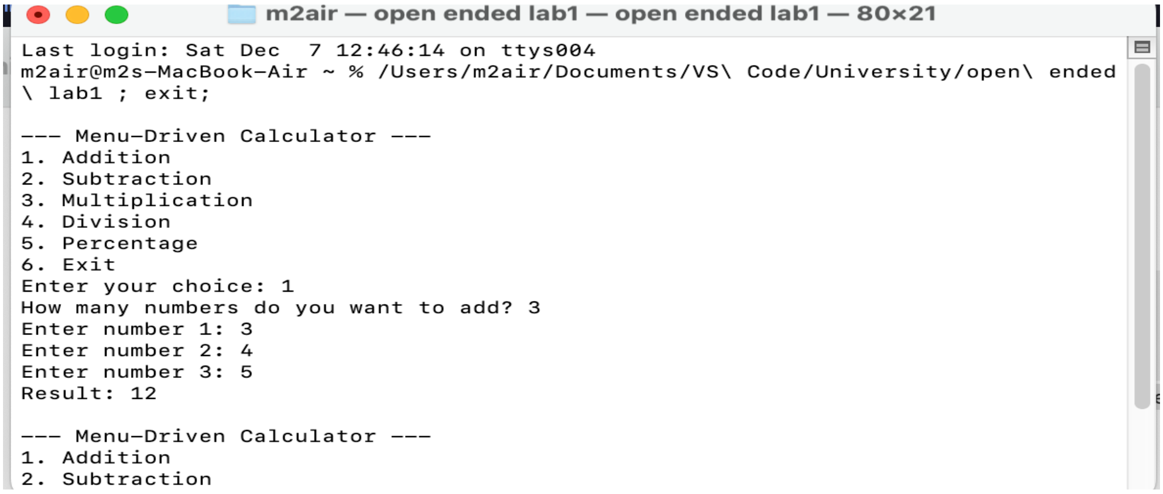
Code

```
#include <stdio.h>
#include <stdlib.h>

// Function prototypes
void addNumbers();
void subtractNumbers();
void multiplyNumbers();
void divideNumbers();
void calculatePercentage();
int getValidNumber();

// Main function
int main() {
    int choice;
    do {
        // Display the menu options
        printf("\n--- Menu-Driven Calculator ---\n");
        printf("1. Addition\n");
        printf("2. Subtraction\n");
        printf("3. Multiplication\n");
        printf("4. Division\n");
        printf("5. Percentage\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        // Perform the operation based on user's choice
        switch (choice) {
            case 1: addNumbers(); break;
            case 2: subtractNumbers(); break;
            case 3: multiplyNumbers(); break;
            case 4: divideNumbers(); break;
            case 5: calculatePercentage(); break;
            case 6: printf("Exiting program.\n"); break;
            default: printf("Invalid choice! Try again.\n");
        }
    } while (choice != 6);

    return 0;
}
```

```
// Function for addition
void addNumbers() {
    int n, num, sum = 0;
    printf("How many numbers do you want to add? ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        printf("Enter number %d: ", i + 1);
        num = getValidNumber();
        sum += num;
    }
    printf("Result: %d\n", sum);
}

// Function for subtraction
void subtractNumbers() {
    int num1, num2;
    printf("Enter first number: ");
    num1 = getValidNumber();
    printf("Enter second number: ");
    num2 = getValidNumber();
    printf("Result: %d\n", num1 - num2);
}

// Function for multiplication
void multiplyNumbers() {
    int n, num, product = 1;
    printf("How many numbers do you want to multiply? ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        printf("Enter number %d: ", i + 1);
        num = getValidNumber();
        product *= num;
    }
    printf("Result: %d\n", product);
}
// Function for division
void divideNumbers() {
    int num1, num2;
    printf("Enter numerator: ");
    num1 = getValidNumber();
    printf("Enter denominator: ");
    num2 = getValidNumber();
    if (num2 == 0) {
        printf("Error: Division by zero is not allowed.\n");
    } else {
        printf("Result: %.2f\n", (float) num1 / num2);
    }
}
// Function for percentage
void calculatePercentage() {
    float num, total;
    printf("Enter the number: ");
    num = getValidNumber();
    printf("Enter the total: ");
    total = getValidNumber();
    if (total == 0) {
        printf("Error: Total cannot be zero.\n");
    } else {
        printf("Percentage: %.2f%%\n", (num / total) * 100);
    }
}

int getValidNumber() {
    int num;
    while (1) {
        if (scanf("%d", &num) == 1) break;
        else {
            printf("Invalid input. Enter a valid number: ");
            while (getchar() != '\n'); // Clear input buffer
        }
    }
    return num;  // Program ends
}
```

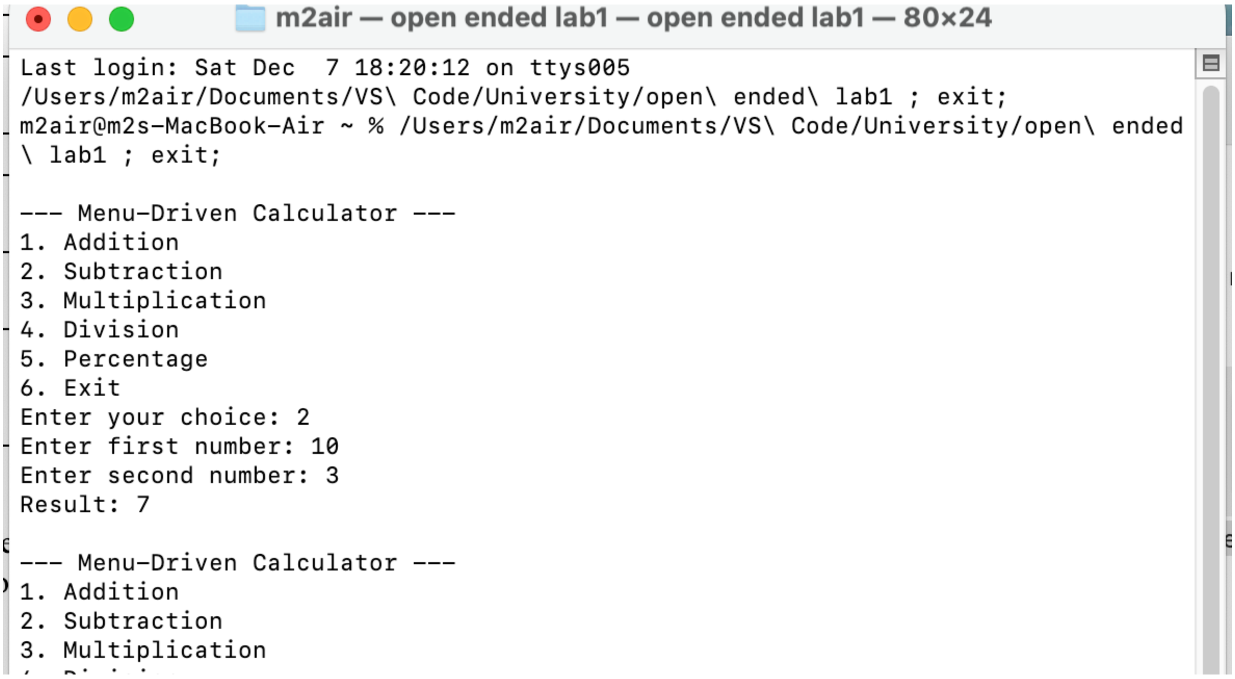
Output Result :

**Addition :**



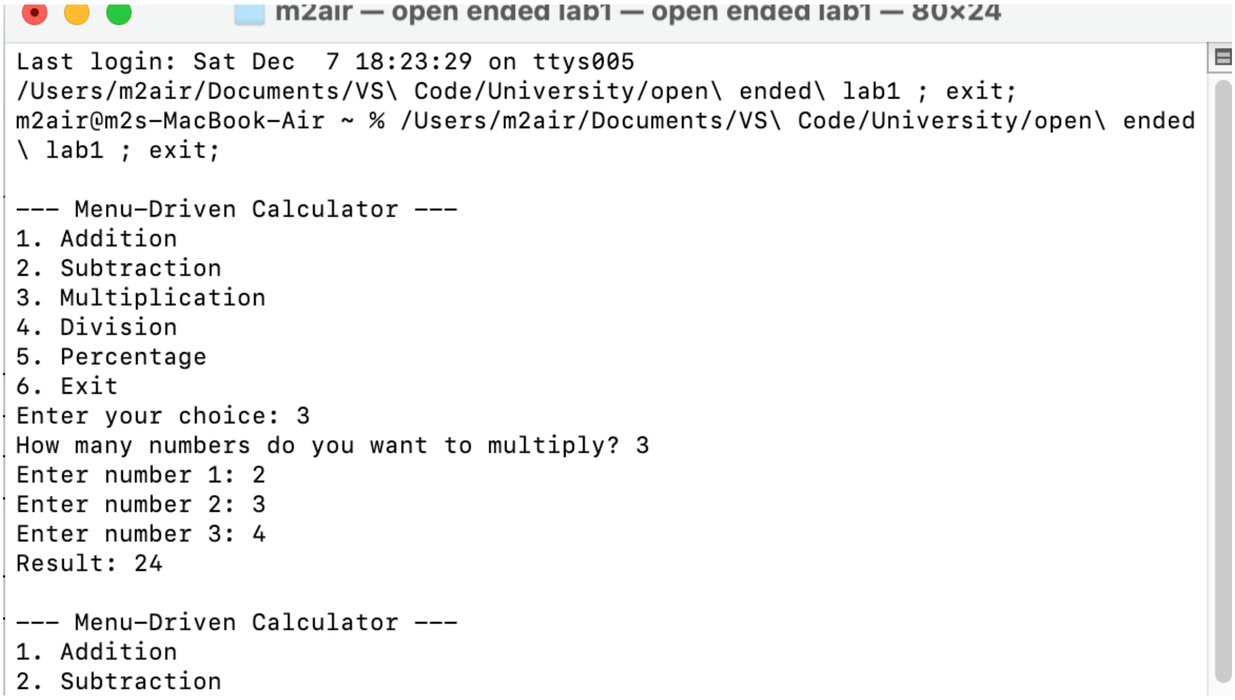
```
Last login: Sat Dec  7 12:46:14 on ttys004
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/open\ ended
\ lab1 ; exit;

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 1
How many numbers do you want to add? 3
Enter number 1: 3
Enter number 2: 4
Enter number 3: 5
Result: 12

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
```

**Subtraction :**



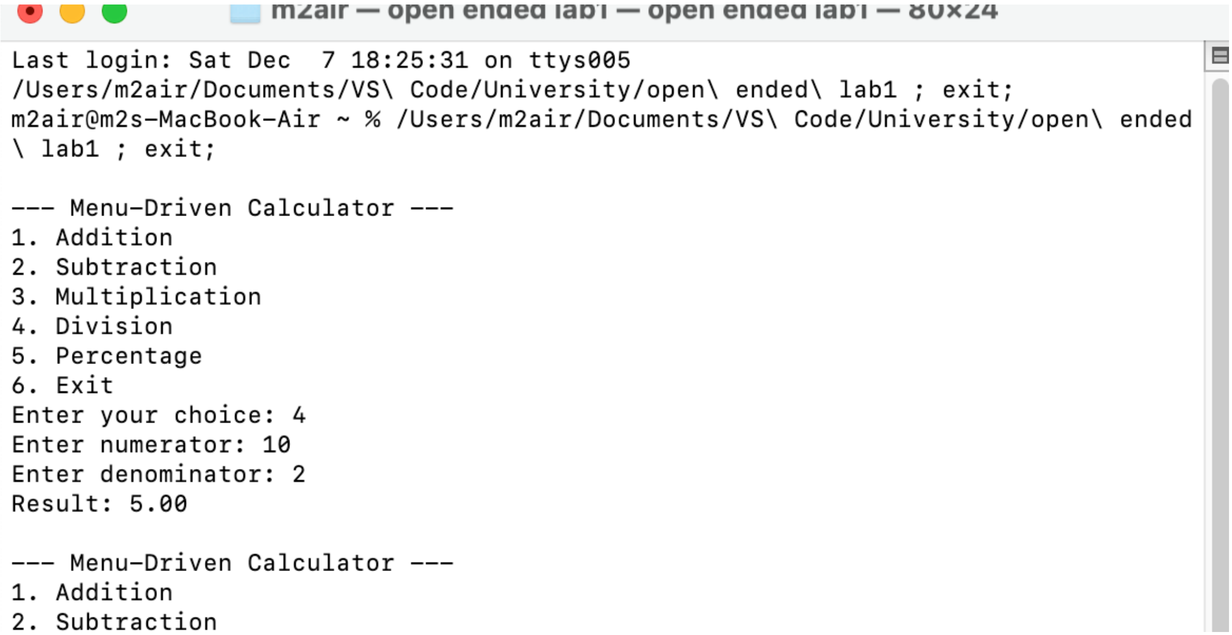
```
Last login: Sat Dec  7 18:20:12 on ttys005
/Users/m2air/Documents/VS\ Code/University/open\ ended\ lab1 ; exit;
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/open\ ended
\ lab1 ; exit;

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 2
Enter first number: 10
Enter second number: 3
Result: 7

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
```

**Multiplication :**



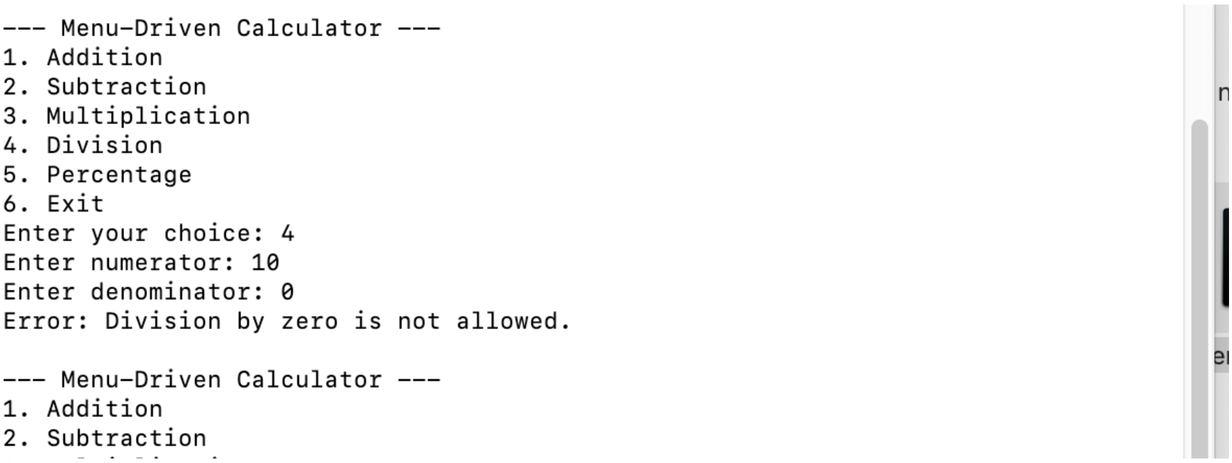
```
Last login: Sat Dec  7 18:23:29 on ttys005
/Users/m2air/Documents/VS\ Code/University/open\ ended\ lab1 ; exit;
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/open\ ended
\ lab1 ; exit;

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 3
How many numbers do you want to multiply? 3
Enter number 1: 2
Enter number 2: 3
Enter number 3: 4
Result: 24

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
```

**Division :**



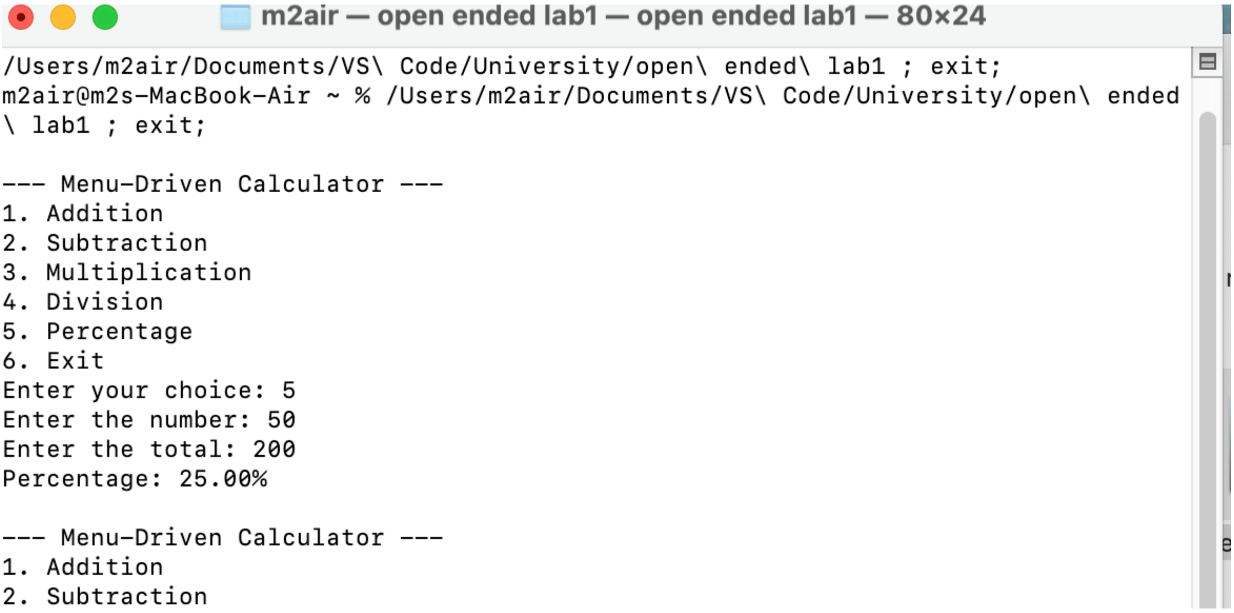
```
Last login: Sat Dec  7 18:25:31 on ttys005
/Users/m2air/Documents/VS\ Code/University/open\ ended\ lab1 ; exit;
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/open\ ended
\ lab1 ; exit;

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 4
Enter numerator: 10
Enter denominator: 2
Result: 5.00

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
```

**Zero Not Allow**



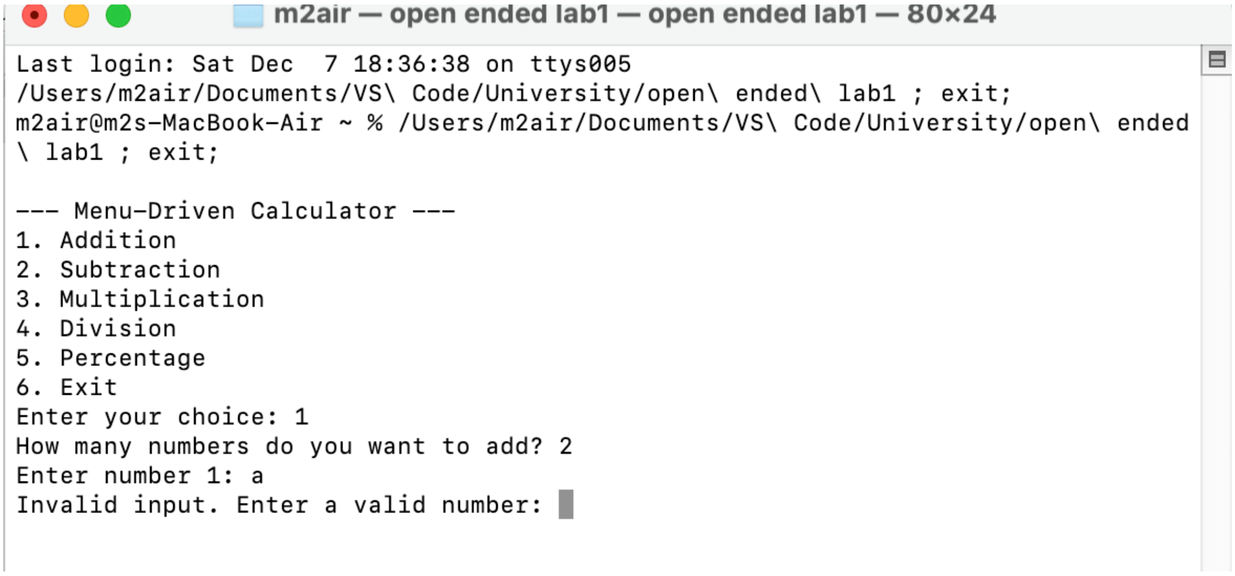
```
--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 4
Enter numerator: 10
Enter denominator: 0
Error: Division by zero is not allowed.

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
```

**Parcentange :**



```
/Users/m2air/Documents/VS\ Code/University/open\ ended\ lab1 ; exit;
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/open\ ended
\ lab1 ; exit;

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 5
Enter the number: 50
Enter the total: 200
Percentage: 25.00%

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
```

**Invalid input: "abc"**



```
Last login: Sat Dec  7 18:36:38 on ttys005
/Users/m2air/Documents/VS\ Code/University/open\ ended\ lab1 ; exit;
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/open\ ended
\ lab1 ; exit;

--- Menu-Driven Calculator ---
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Percentage
6. Exit
Enter your choice: 1
How many numbers do you want to add? 2
Enter number 1: a
Invalid input. Enter a valid number: 
```

4. Submit a well-documented lab report including requirement analysis, algorithms, code, and test results.

Answer:

1. **Introduction:**
   The goal of this experiment is to develop a menu-driven calculator using the C programming language. The program provides a user-friendly interface to perform basic mathematical operations such as addition, subtraction, multiplication, division, and percentage calculation. It emphasizes robust input validation and error handling, ensuring a seamless and error-free user experience.

2. **Requirement Analysis:**
   The calculator performs basic mathematical operations:

   **Operations Supported:**

   1.Addition, subtraction, multiplication, division (with error handling), and percentage calculation.

   2. Option to exit the program.

   **Validation Rules:**

   1. Inputs must be valid numbers.
   2. Division by zero is not allowed.
   3. Proper error messages for invalid inputs

3. **Algorithms:**
   a) Addition: Sum up numbers provided by the user.
   b) Subtraction: Subtract one number from another.
   c) Multiplication: Multiply multiple numbers sequentially.
   d) Division: Divide two numbers, ensuring the divisor isn't zero.
   e) Percentage: Calculate (number / total) * 100 with validation.
   f) Input Validation: Ensure only numbers are accepted.

4. **Code Implementation:**
   Refer to the provided annotated C code, which implements the features as discussed.

5**. Testing Results:**

| Test Case | Input | Expected Output | Expected Output |
|---|---|---|---|
| 1 | Addition: 3, 4, 5 | Result: 12 | Pass |
| 2 | Subtraction: 10, 3 | Result: 7 | Pass |
| 3 | Multiplication: 2, 3, 4 | Result: 24 | Pass |
| 4 | Division: 10, 2 | Result: 5 | Pass |
| 5 | Division: 10, 0 | Error: Division by zero is not allowed | Pass |

| | | | |
|---|---|---|---|
| 6 | Percentage: 50, 200 | Result: 25% | Pass |
| 7 | invalid input: "abc" | Error: Invalid input. Please enter a valid number | Pass |

**Conclusion:**
The program successfully implements a menu-driven calculator capable of performing various mathematical operations with robust input validation. It meets the requirements for functionality, input handling, and error prevention.