



Lab Report

Faculty of Engineering and Technology

Department of Information and Communication Engineering

Course Code : **ICE-4206**

Course Title : **Neural Network Sessional**

Submitted By :

Md. Sha Alam

Roll : **190610**

Registration No : **1065335**

Session : **2018-19**

Department of **ICE** , **PUST**

Submitted To :

Dr. Md. Imran Hossain

Associate Professor,

Department of **ICE**,

PUST

Submission Date: 15-9-2024

signature

Source Code:

```
#include <stdio.h>
int main() {
    int a, b;
    char operator, separator;
    double result;

    // Array to store each expression and their corresponding results
    char expressions[100][50]; // To store the formatted expressions with results
    int expr_count = 0; // Count of expressions

    printf("Enter a number,operand and then another number.After that enter equal sign or
    semicolon :\n");

    while (1) {
        // Input each expression in the format a operator b separator

        scanf("%d %c %d %c", &a, &operator, &b, &separator);

        // Perform the operation based on the operator
        switch (operator) {
            case '+':
                result = a + b;
                break;
            case '-':
                result = a - b;
                break;
            case '*':
                result = a * b;
                break;
            case '/':
                if (b != 0) {
                    result = (double)a / b;
                } else {
                    printf("Error: Division by zero!\n");
                    return 1;
                }
                break;

            case '%':
                if (b != 0) {
                    result = a % b;
                } else {
```

```

        printf("Error: Modulo by zero!\n");
        return 1;
    }
    break;
default:
    printf("Invalid operator!\n");
    return 1;
}

// Format and store the expression result
if (separator == ';') {
    sprintf(expressions[expr_count], "%d %c %d = %.2f", a, operator, b, result); //
Store the expression with result
} else if (separator == '=') {
    sprintf(expressions[expr_count], "%d %c %d = %.2f", a, operator, b, result); //
Store the final expression with result
    expr_count++; // Make sure to count the final expression
    break; // Exit the loop after '=' is encountered
} else {
    printf("Invalid input format. Please use ';' or '=' to separate expressions.\n");
    return 1;
}

expr_count++; // Increment the expression count
}

// Output all expressions at once after '=' is encountered
for (int i = 0; i < expr_count; i++) {
    printf("%s\n", expressions[i]);
}

return 0;
}

```

Output:

Enter a number,operand and then another number.After that enter equal sign or semicolon
:
8 + 5;8*2=
8 + 5 = 13.00
8 * 2 = 16.00

Source Code:

```
def calculation(num1, op, num2):
    if op == "+":
        return num1 + num2
    elif op == "-":
        return num1 - num2
    elif op == "*":
        return num2 * num1
    elif op == "/":
        return num1 / num2
    elif op == "%":
        return num1 % num2
    else:
        print("Please Enter Valid Operator(+,-,/,*,%)")

n = int(input("Please Enter n: "))
elements = input(
    f"Please Enter {n * 2} Elements space by space and at last please enter a operator(+,-,*,/,%): ").split(" ")
op = elements[-1]
elements = list(map(int, elements[:-1]))
for i in range(0, 2 * n, 2):
    cal = calculation(elements[i], op, elements[i + 1])
    print(cal, end=" ")
print()
```

Output:

```
Please Enter n: 3
Please Enter 6 Elements space by space and at last please enter a operator(+,-,*,/,%): 4 5
6 4 5 5 +
9 10 10
```

Source Code:

```
string = input("Please Enter a Number or string: ")
revString=string[::-1]
if string==revString:
    print("The Number or String is Palindrome")
else:
    print("The Number or String is not Palindrome")
```

Output:

```
Please Enter a Number or string: 12521
The Number or String is Palindrome
```

Source Code:

```
n=int(input("Please Enter a number: "))
fact=1
for i in range(1,n+1):
    prev=fact
    fact = i*fact
    # To verify the factorial
    print(f"In {i} iteration factorial is {prev} X {i} = {fact}")
print(f"Factorial Of {n} is {fact}")
```

Output:

```
Please Enter a number: 6
In 1 iteration factorial is 1 X 1 = 1
In 2 iteration factorial is 1 X 2 = 2
In 3 iteration factorial is 2 X 3 = 6
In 4 iteration factorial is 6 X 4 = 24
In 5 iteration factorial is 24 X 5 = 120
In 6 iteration factorial is 120 X 6 = 720
Factorial Of 6 is 720
```

Source Code:

```
#include <stdio.h>

// Function to calculate the sum of the array
int calculateSum(int arr[], int size) {
    int sum = 0;
    int i = 0;
    do {
        sum += arr[i];
        i++;
    } while (i < size);
    return sum;
}

// Function to calculate the average of the array
double calculateAverage(int arr[], int size) {
    int sum = calculateSum(arr, size); // Get the sum using the calculateSum function
    return (double)sum / size;
}

int main() {
    int n;

    // Input the size of the array
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    if (n <= 0) {
        printf("Array size must be a positive integer.\n");
        return 1;
    }

    int arr[n]; // Declare an array of size n

    // Input elements of the array
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Calculate sum and average
    int sum = calculateSum(arr, n);
    double average = calculateAverage(arr, n);
```

```
// Display results
printf("Sum of the array elements: %d\n", sum);
printf("Average of the array elements: %.2f\n", average);

return 0;
}
```

Output:

Enter the number of elements: 4

Enter 4 elements:

10

5

15

20

Sum of the array elements: 50

Average of the array elements: 12.50

Source Code:

```
import java.io.*;
public class lab7 {
    public static void main(String[] args) {
        DataInputStream inputStream=null;
        // Example 1: ClassNotFoundException
        try {
            Class.forName("NonExistentClass");
        } catch (ClassNotFoundException e) {
            System.out.println("ClassNotFoundException occurred: " + e.getMessage());
        }
        // Example 2: EOFException
        try {
            inputStream = new DataInputStream(new FileInputStream("nonexistent.txt"));
            while (true) {
                try {
                    char c = (char)inputStream.readByte();
                    System.out.print(c);
                } catch (EOFException eof) {
                    System.out.println("\nEnd of file reached.");
                    break;
                } catch (IOException ioe) {
                    ioe.printStackTrace();
                    break;
                }
            }
        } catch (FileNotFoundException fnfe) {
            System.out.println("File not found: nonexistent.txt");
        } finally {
            try {
                inputStream.close();
            } catch (IOException ioe) {
                ioe.printStackTrace();
            }
        }
    }
}
```

Output:

ClassNotFoundException occurred: NonExistentClass
System Analysis and Software Testing
End of file reached.

Source Code:

```
def calculation(num1, op, num2):
    if op == "+":
        return num1 + num2
    elif op == "-":
        return num1 - num2
    elif op == "*":
        return num2 * num1
    elif op == "/":
        return num1 / num2
    else:
        print("Please Enter Valid Operator(+,-,/,*)")

with open('input.txt', 'r') as file:
    input_data = file.read()

elements = list(map(int, input_data.replace(',', ' ').split()))
n = len(elements)

for i in range(0, n, 2):
    add = calculation(elements[i], "+", elements[i + 1])
    sub = calculation(elements[i], "-", elements[i + 1])
    mul = calculation(elements[i], "*", elements[i + 1])
    div = calculation(elements[i], "/", elements[i + 1])
    result = [add, sub, mul, div]
    with open('output.txt', 'a') as file:
        file.write(f"Case: {i // 2 + 1}: " + ' '.join(map(str, result)) + "\n")
print("Successfully Created File with Required Calculations")
print()
```

Output:

Successfully Created File with Required Calculations

Input text :

5 5 9 8

Output text:

Case: 1: 10 0 25 1.0

Case: 2: 17 1 72 1.125