ASSIGMENT - 2

1.Analysis of the Problem:

Inputs:

The equation to be solved dxdy +2xcot(y)=0

Initial values: x=π and y=x

Process:

Solve the differential equation to find the particular solution for y in terms of x.

Write a C program that takes an input value of x and computes the corresponding value of y using the solved differential equation.

Outputs:

Computed value of y for the given x.

Algorithm:

Steps:

Read the input value for x.

Define the function dydx to compute the derivative dxdy using the differential equation +2dxdy+2xcot(y)=0.

Use an iterative method (such as Euler's method) to solve the differential equation numerically and find the value of y for the given x.

Display the computed value of y.

Pseudocode:

1. Read input value for x.

2. Define function dydx(y, x) to calculate the derivative dy/dx using the differential equation.

3. Initialize variables: y = π (given initial value), step_size, and the desired x.

4. Use an iterative method (e.g., Euler's method) to find y for the given x:

    a. while (current_x < desired_x):

        i. Calculate dydx_val = dydx(y, current_x).

        ii. Update y = y + dydx_val * step_size.

        iii. Update current_x = current_x + step_size.

5. Display the computed value of y.

```
#include <stdio.h>

#include <math.h>

// Function to compute the derivative dy/dx using the differential equation

double dydx(double y, double x) {
```

ASSIGMENT - 2

```
    return -2 * x * (1 / tan(y)); // dy/dx = -2x * cot(y)
}
int main() {
    double x, y = M_PI; // Given initial value for y when x = π
    double desired_x, step_size = 0.001; // Define the desired x and step size
    printf("Enter the value of x: ");
    scanf("%lf", &desired_x); // Read the input value for x
    double current_x = M_PI; // Starting from x = π
    // Use Euler's method to solve the differential equation and find y for the given x
    while (current_x < desired_x) {
        double dydx_val = dydx(y, current_x); // Calculate dy/dx
        y = y + dydx_val * step_size; // Update y using Euler's method
        current_x += step_size; // Update x for the next iteration
    }
    printf("For x = %.4lf, y = %.4lf\n", desired_x, y); // Display the computed value of y
    return 0;
}
```

Displaying Inputs and Desired Output:

Inputs taken:

Value of x

Desired output:

Computed value of y for the given x

Testing:

Test the program with various values of x to observe the corresponding computed values of y. For instance:

Input: x=3

Input: x=5

Input: x=7

Input: x=10

ASSIGMENT - 2

2.Analysis of the Problem:

Inputs:

The given first-order differential equation: dydx +xtan(y)=cos 2 (x)

Variable values: x and y

Process:

Solve the differential equation to find the value of the constant C in the general solution y= cos(x)cos2 (x) −x2 +C.

Write a C program to compute the constant C for any given values of x and y.

Outputs:

Computed value of the constant C for the provided x and y.

Algorithm:

Steps:

Read the input values for x and y.

Define a function calculate_C to compute the constant C using the given differential equation.

Implement an algorithm to solve the equation for C based on the provided x and y values.

Display the computed value of C.

Pseudocode:

1. Read input values for x and y.
2. Define function calculate_C(x, y) to compute the constant C using the differential equation.
3. Calculate C using the provided x and y:
   a. Calculate cos_squared_x = cos(x) * cos(x).
   b. Calculate C = (cos_squared_x / cos(x)) - (x * x) - (y * tan(y)).
4. Display the computed value of C.

```
#include <stdio.h>

#include <math.h>

double calculate_C(double x, double y) {

    double cos_squared_x = cos(x) * cos(x);

    double C = (cos_squared_x / cos(x)) - (x * x) - (y * tan(y));

    return C;

}
```

ASSIGMENT - 2

```
int main() {
    double x, y;
    printf("Enter the value of x: ");
    scanf("%lf", &x);
    printf("Enter the value of y: ");
    scanf("%lf", &y);
    double C = calculate_C(x, y);
    printf("For x = %.4lf and y = %.4lf, the constant C = %.4lf\n", x, y, C);
    return 0;
}
```

Displaying Inputs and Desired Output:

Inputs taken:

Values of x and y

Desired output:

Computed value of the constant C for the given x and y

Testing:

Test the program with various values of x and y to compute the corresponding constant

C. For instance:

Input: x=1.5, y=0.8

Input: x=2.3, y=1.1

Input: x=0.7, y=0.5

Let's break down the problem and outline the necessary steps to create a C program to compute the flow of current in the circuit.

3.Analysis of the Problem:

Inputs:

Values of constants: E (electromotive force), R (resistance), L (self-inductance), and t (time)

Process:

Solve the given differential equation to find the flow of current i in the circuit.

Write a C program to compute the current flow i for the provided values of E, R, L, and t.

ASSIGMENT - 2

Outputs:

Computed value of the current flow i in the circuit.

Algorithm:

Steps:

Read the input values for E, R, L, and t.

Define a function calculate_current_flow to compute the flow of current using the given differential equation.

Implement an algorithm to solve the differential equation numerically for the flow of current i based on the provided constants and time t.

Display the computed value of the current flow i.

Pseudocode:

1. Read input values for E, R, L, and t.

2. Define function calculate_current_flow(E, R, L, t) to solve the differential equation and compute the current flow i.

3. Use a numerical method (e.g., Euler's method) to solve the differential equation:

   a. Initialize variables: i = 0 (initial current)

   b. Define a step size

   c. while (time < t):

      i. Calculate di_dt = (E - R * i) / L

      ii. Update i = i + di_dt * step_size

      iii. Update time = time + step_size

4. Display the computed value of the current flow i.

```
#include <stdio.h>


double calculate_current_flow(double E, double R, double L, double t) {
    double i = 0; // Initial current
    double step_size = 0.001; // Define step size for numerical integration
    double time = 0;


    // Use Euler's method to solve the differential equation and compute current flow i
    while (time < t) {
        double di_dt = (E - R * i) / L; // Calculate di/dt
```

ASSIGMENT - 2

```
        i = i + di_dt * step_size; // Update current i using Euler's method
        time += step_size; // Update time for the next iteration
    }
    return i;
}
int main() {
    double E, R, L, t;
    printf("Enter the value of E: ");
    scanf("%lf", &E);
    printf("Enter the value of R: ");
    scanf("%lf", &R);
    printf("Enter the value of L: ");
    scanf("%lf", &L);
    printf("Enter the value of t: ");
    scanf("%lf", &t);
    double current_flow = calculate_current_flow(E, R, L, t);
    printf("For E=%.4lf, R=%.4lf, L=%.4lf, and t=%.4lf, the current flow i=%.4lf\n", E, R, L, t,
current_flow);
    return 0;
}
```

Displaying Inputs and Desired Output:

Inputs taken:

Values of E, R, L, and t

Desired output:

Computed value of the current flow i for the given constants and time t

Testing:

Test the program with various values of E, R, L, and t to compute the corresponding current flow i.

For instance:

Input: E=12, R=4, L=2, t=5

Input: E=15, R=3, L=4, t=8

Input: E=10, R=5, L=3, t=10

ASSIGMENT - 2

4.Analysis of the Problem:

Inputs:

Values of constants: E (constant), p, L, and t

Process:

Solve the given differential equation to find the flow of current i in the circuit.

Utilize the resonance condition LC1 =p 2

to simplify the differential equation.

Write a C program to compute the current flow i for the provided values of E, p, L, and t.

Outputs:

Computed value of the current flow i in the circuit.

Algorithm:

Steps:

Read the input values for E, p, L, and t.

Use the resonance condition to simplify the differential equation and obtain the expression for the current i.

Implement an algorithm to compute the current i using the derived expression.

Display the computed value of the current flow i.

Pseudocode:

1. Read input values for E, p, L, and t.
2. Calculate the frequency of the circuit using resonance condition: freq = 1 / (2 * pi * sqrt(L * C))
3. Compute the current flow i using the given expression:
   a. Calculate the sine function with frequency freq and time t.
   b. Multiply the result by E, L, and the sine function to get the current i.
4. Display the computed value of the current flow i.

```
#include <stdio.h>

#include <math.h>

double calculate_current_flow(double E, double p, double L, double t) {

    double freq = 1.0 / (2.0 * M_PI * sqrt(L * pow(p, 2))); // Calculate the frequency

    // Compute the current flow i using the given expression

    double i = E * L * sin(2 * M_PI * freq * t);

    return i;
```

```
}
int main() {
    double E, p, L, t;
    printf("Enter the value of E: ");
    scanf("%lf", &E);
    printf("Enter the value of p: ");
    scanf("%lf", &p);
    printf("Enter the value of L: ");
    scanf("%lf", &L);
    printf("Enter the value of t: ");
    scanf("%lf", &t);
    double current_flow = calculate_current_flow(E, p, L, t);
    printf("For E=%.4lf, p=%.4lf, L=%.4lf, and t=%.4lf, the current flow i=%.4lf\n", E, p, L, t, current_flow);
    return 0;
}
```

Displaying Inputs and Desired Output:

Inputs taken:

Values of E, p, L, and t

Desired output:

Computed value of the current flow i for the given constants and time t

Testing:

Test the program with various values of E,

p, L, and t to compute the corresponding current flow i.

For instance:

Input: E=10, p=2, L=3, t=0.5

Input: E=12, p=1.5, L=4, t=1

Input: E=15, p=2.5, L=2.5, t=0.8