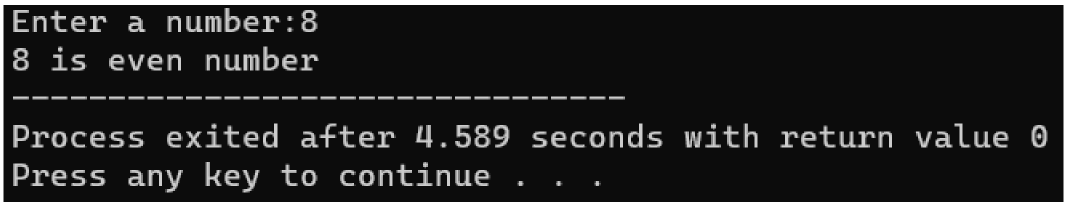
Embedded C Programming – Laboratory (L33+L34)

Experiment 2 – Programs on Condition and Control Statements

Task 1.01:

```
#include<stdio.h>

int main(){           // if-statement to check an even
number

    int number=0;      // Declare an integer variable
'number' and initialize it to 0

    printf("Enter a number:");  // Display a prompt to the
user

    scanf("%d",&number);        // Read an integer from
the user and store it in the 'number' variable

     if(number % 2 == 0){         // Check if the number
is even using the modulo operator

      printf("%d is an even number", number);  // Display
a message if the number is even

    }

   return 0;

}
```

Output:

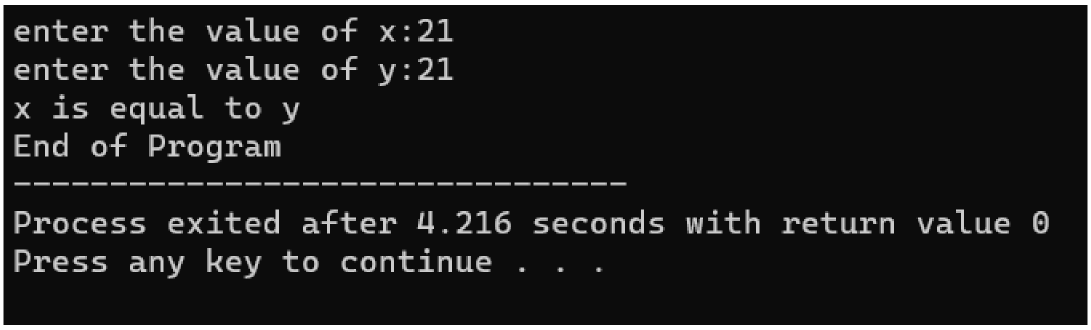


```
Enter a number:8
8 is even number
--------------------------------
Process exited after 4.589 seconds with return value 0
Press any key to continue . . .
```

Task 1.02:

```
#include <stdio.h>

int main()

{//multiple if statements

    int x, y;
    printf("enter the value of x:");
    scanf("%d", &x);
    printf("enter the value of y:");
    scanf("%d", &y);
    if (x>y)
    {
     printf("x is greater than y\n");
    }
    if (x<y)
    {
     printf("x is less than y\n");
    }
    if (x==y)
    {
     printf("x is equal to y\n");
    }
    printf("End of Program");
    return 0;
}
```

Output:



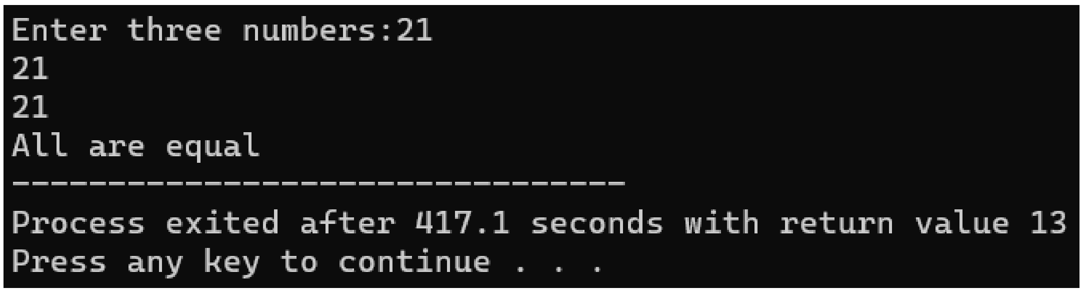
```
enter the value of x:21
enter the value of y:21
x is equal to y
End of Program
--------------------------------
Process exited after 4.216 seconds with return value 0
Press any key to continue . . .
```

Task 1.03:

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

```c
int main()
{  //program to find largest of three numbers
    int a, b, c;
     printf("Enter three numbers:");
    scanf("%d %d %d",&a,&b,&c);
    if(a>b && a>c)
    {
        printf("%d is largest",a);
    }
    if(b>a && b > c)
    {
        printf("%d is largest",b);
    }
    if(c>a && c>b)
    {
        printf("%d is largest",c);
    }
    if(a == b && a == c)
    {
        printf("All are equal");
    }
}
```

Output:

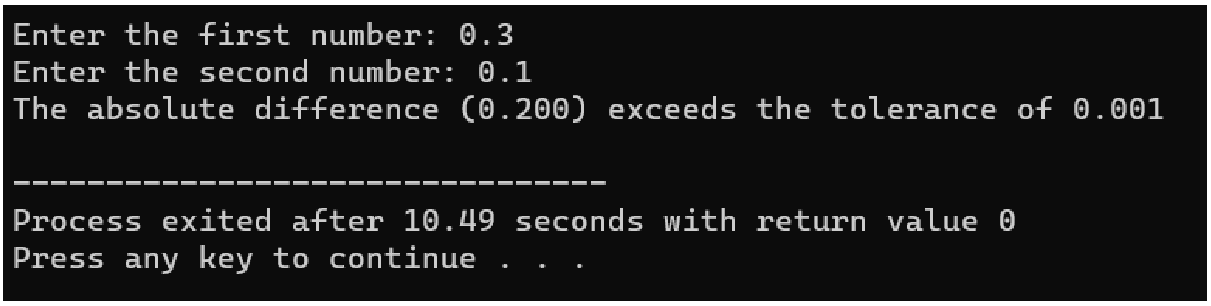


```
Enter three numbers:21
21
21
All are equal
--------------------------------
Process exited after 417.1 seconds with return value 13
Press any key to continue . . .
```

Task 1.04:

```
#include <stdio.h>

#include <math.h>

//Program on difference and comparison using if-else

int main() {

    // Variables

    double num1, num2, difference;

    double tolerance = 0.001;

    // Getting user input

    printf("Enter the first number: ");

    scanf("%lf", &num1);

    printf("Enter the second number: ");

    scanf("%lf", &num2);

    // Calculating absolute difference

    difference = fabs(num1 - num2);

    // Checking if the difference is within the tolerance

    if (difference < tolerance) {

        printf("The absolute difference (%.3f) is within
the tolerance of %.3f\n", difference, tolerance);

    } else {

        printf("The absolute difference (%.3f) exceeds the
tolerance of %.3f\n", difference, tolerance);

    }

    return 0;

}
```

Output:

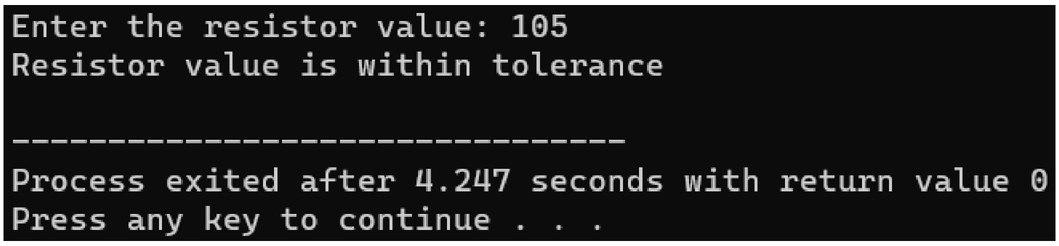


```
Enter the first number: 0.3
Enter the second number: 0.1
The absolute difference (0.200) exceeds the tolerance of 0.001

--------------------------------
Process exited after 10.49 seconds with return value 0
Press any key to continue . . .
```

Task 1.05:

```
#include <stdio.h>
int main() {/*if-else statement
    Resistor value range and tolerance*/
    float nominalValue = 100.0;//in ohms
    float tolerance = 5.0;
    // User-input resistor value
    float userResistor;
    // Getting user input
    printf("Enter the resistor value: ");
    scanf("%f", &userResistor);
    // Checking if the resistor value is within the
specified tolerance (95-105 ohms)
    if (userResistor >= (nominalValue - tolerance) &&
userResistor <= (nominalValue + tolerance)) {
        printf("Resistor value is within tolerance\n");
    } else {
        printf("Resistor value is outside tolerance\n");
    }
    return 0;
}
```

Output:

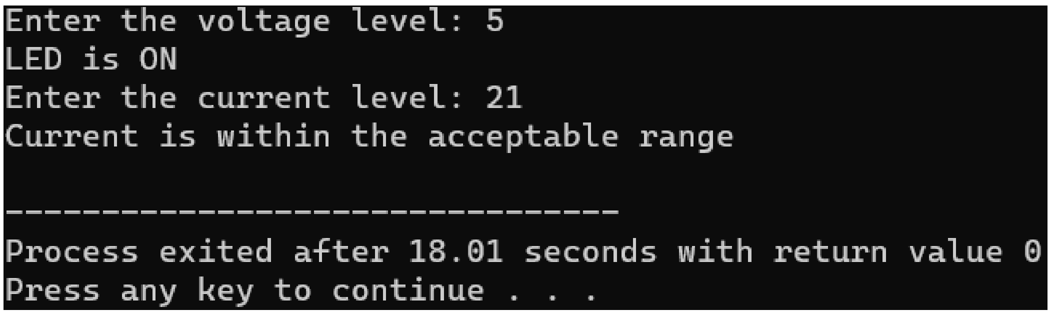

```
Enter the resistor value: 105
Resistor value is within tolerance

--------------------------------
Process exited after 4.247 seconds with return value 0
Press any key to continue . . .
```

Task 1.06:

Write a program to check the voltage and current. If the voltage is greater than or equal to threshold voltage (3v), the LED should be ‘ON’. If this condition is satisfied, check whether the current is withing the specified range (10-30 A), and print that the current is within the specified range. Otherwise, if the voltage is less than threshold voltage, mark the LED as ‘OFF’. (Hint: Use if-else ladder statement)

```
#include <stdio.h>

int main() {//if-else ladder statement

    // Voltage threshold for LED activation

    float voltageThreshold = 3.0;

    // Current range for LED activation

    float minCurrent = 10.0;

    float maxCurrent = 30.0;

    // User-input voltage and current

    float userVoltage, userCurrent;

    // Getting user input

    printf("Enter the voltage level: ");

    scanf("%f", &userVoltage);

    // Checking if the voltage is above the threshold

    if (userVoltage >= voltageThreshold) {

        printf("LED is ON\n");
        // If the voltage is above the threshold, check
the current range
        printf("Enter the current level: ");
        scanf("%f", &userCurrent);

        // Nested if-else to check if the current is
within the specified range
        if (userCurrent >= minCurrent && userCurrent <=
maxCurrent) {
            printf("Current is within the acceptable
range\n");
        } else {
            printf("Current is outside the acceptable
range\n");
        }
    } else {
        printf("LED is OFF\n");
    }
    return 0;
}
```

Output:



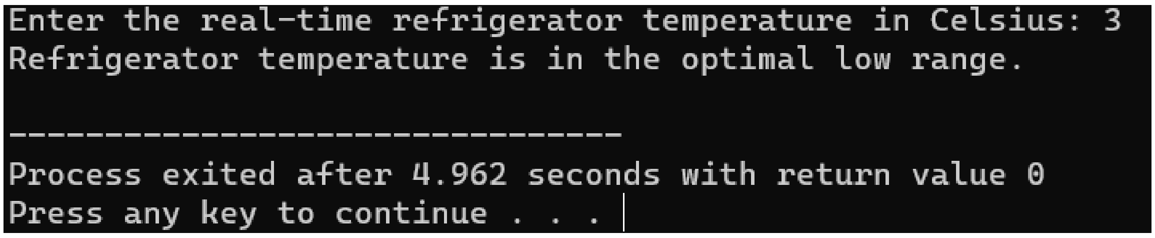
```
Enter the voltage level: 5
LED is ON
Enter the current level: 21
Current is within the acceptable range

--------------------------------
Process exited after 18.01 seconds with return value 0
Press any key to continue . . .
```

Task 1.07:

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

```
int main() {/*else-if ladder
    Temperature thresholds for different conditions in
Celsius*/
    float lowTempThreshold = 2.0;
    float mediumTempThreshold = 8.0;
    float highTempThreshold = 15.0;
    // User-input real-time refrigerator temperature
    float refrigeratorTemperature;
    // Getting user input using standard functions
    printf("Enter the real-time refrigerator temperature
in Celsius: ");
    scanf("%f", &refrigeratorTemperature);
    // Using else-if ladder to check temperature
conditions
    if (refrigeratorTemperature < lowTempThreshold) {
        printf("Refrigerator temperature is too low.
Adjust the settings.\n");
    } else if (refrigeratorTemperature <
mediumTempThreshold) {
        printf("Refrigerator temperature is in the optimal
low range.\n");
    } else if (refrigeratorTemperature <
highTempThreshold) {
        printf("Refrigerator temperature is in the optimal
medium range.\n");
    } else {
        printf("Refrigerator temperature is too high.
Adjust the settings.\n");
    }
    // Wait for a key press before exiting
    getchar();
    return 0;
}
```

Output:

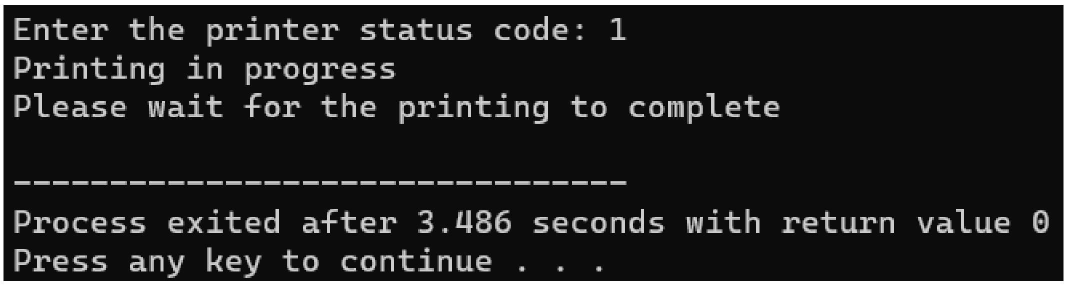


```
Enter the real-time refrigerator temperature in Celsius: 3
Refrigerator temperature is in the optimal low range.

--------------------------------
Process exited after 4.962 seconds with return value 0
Press any key to continue . . .
```

Task 1.08:

```
#include <stdio.h>
#include <stdbool.h> // Include the header file for
boolean type
int main() {
    // Printer status codes
    int printerStatusCode;
    // Boolean variable to track printing status
    bool isPrinting = false;
    // Getting user input
    printf("Enter the printer status code: ");//code can
be 1,2,3,0 (valid) or other (invalid)
    scanf("%d", &printerStatusCode);
    // Using else-if ladder to check printer status
    if (printerStatusCode == 0) {
        printf("Printer is idle\n");
        isPrinting = false;
    } else if (printerStatusCode == 1) {
        printf("Printing in progress\n");
        isPrinting = true;
    } else if (printerStatusCode == 2) {
        printf("Paper jam detected\n");
        isPrinting = false;
    } else if (printerStatusCode == 3) {
        printf("Out of paper\n");
        isPrinting = false;
    } else {
        printf("Unknown printer status code\n");
        isPrinting = false;
    }
    // Additional action based on the printing status
    if (isPrinting) {
        printf("Please wait for the printing to 
complete\n");
    } else {
        printf("You can proceed with other tasks\n");
    }
    return 0;
}
```

Output:



```
Enter the printer status code: 1
Printing in progress
Please wait for the printing to complete

--------------------------------
Process exited after 3.486 seconds with return value 0
Press any key to continue . . .
```

Take home tasks:

Task 1.09:

Assume the threshold voltage as 5v. Turn the circuit ‘ON’ if the user voltage is greater than threshold voltage. Write a C program for the above problem using if-else statement.

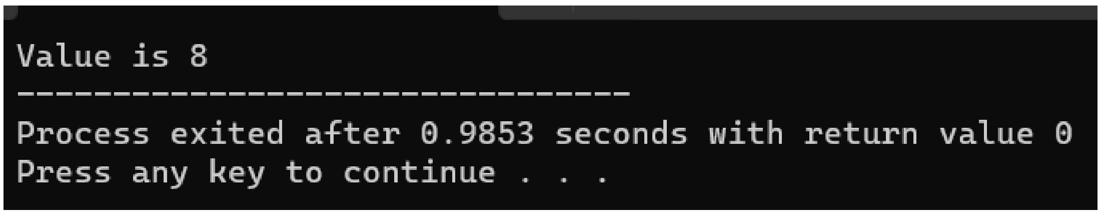
Task 1.10:

A washing machine has three controls: normal wash (control=1), heavy wash (control=2), and quick wash (control=3). If the control is 0, the machine is idle, and print as idle. For any controls other than the above, washing machine should give an unknown control message. Use else-if ladder statement and unsigned int datatype to write a C program for the above problem.

Task 1.11:

```c
#include <stdio.h>
    int main() {
        int num = 8;
        switch (num) {
            case 7:
                printf("Value is 7");
                break;
            case 8:
                printf("Value is 8");
                break;
            case 9:
                printf("Value is 9");
                break;
            default:
                printf("Out of range");
                break;
        }
        return 0;
    }
```

Output:

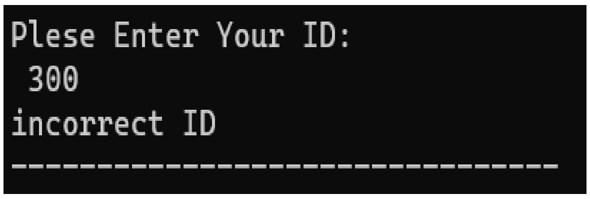


```
Value is 8
--------------------------------
Process exited after 0.9853 seconds with return value 0
Press any key to continue . . .
```

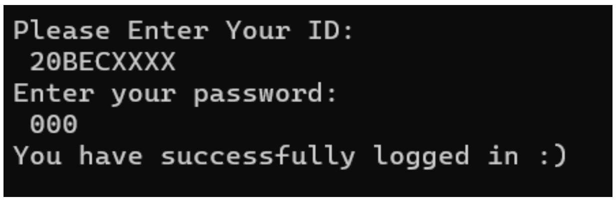
Task 1.12:

```
#include <stdio.h>
#include <string.h>
int main() {
    char ID[10];  // Assuming the ID can be up to 9 characters (plus null terminator)
    int password = 000;
    printf("Please Enter Your ID:\n ");
    scanf("%s", ID);
    // Compare the entered ID with the desired pattern
    if (strcmp(ID, "20BECXXXX") == 0) {
     /*strcmp lexicographically compares the string value
     if equal - returns 0 value;
     if less - returns -ve value; if high - returens +ve value;*/
        printf("Enter your password:\n ");
        scanf("%d", &password);
        switch (password) {
            case 000:
                printf("You have successfully logged in :)\n");
                break;
            default:
                printf("Incorrect password\n");
                break;
        }
    } else {
        printf("Incorrect ID\n");
    }
```

Output:



```
Plese Enter Your ID:
 300
incorrect ID
--------------------------------
```



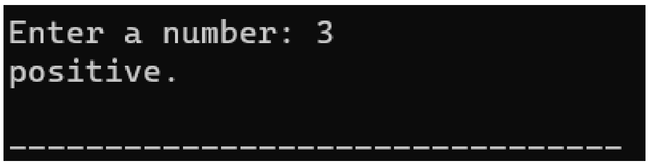
```
Please Enter Your ID:
 20BECXXXX
Enter your password:
 000
You have successfully logged in :)
```

Task 1.13:

//Program using '?:'conditional operator

```
#include <stdio.h>
int main() {
    int No;
    printf("Enter a number: ");
    scanf("%d", &No);
    // Using the ? : operator to determine if the number
is positive or negative
    (No >= 0) ? printf("positive\n") :
printf("negative\n");
    return 0;
}
```

Output:



```
Enter a number: 3
positive.

--------------------------------
```

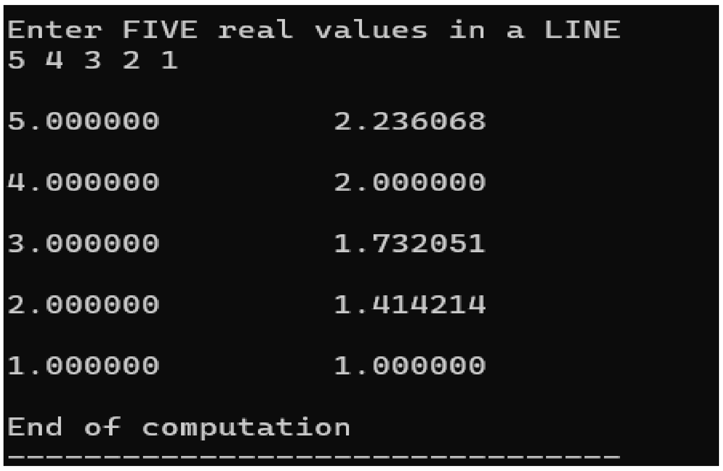
Task 1.14:

//Program using 'goto' statement

```
#include <math.h>
main()
{
     double x, y;
```

```
	int count;
	count = 1;
	printf("Enter FIVE real values in a LINE \n");
read:
	scanf("%lf", &x);
	printf("\n");
	if (x < 0)
		printf("Value - %d is negative\n",count);
	else
	{
		y = sqrt(x);
		printf("%lf\t %lf\n", x, y);
	}
count = count + 1;
	if (count <= 5)
	goto read;/*it goes to label 'read' until count is 5,
	for above 5-it moves to next statement*/
	printf("\nEnd of computation");
}
```

Output:



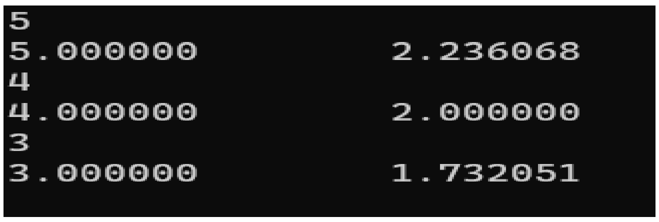
```
Enter FIVE real values in a LINE
5 4 3 2 1

5.000000         2.236068

4.000000         2.000000

3.000000         1.732051

2.000000         1.414214

1.000000         1.000000

End of computation
--------------------------------
```

Task 1.15:

//infinite loop

```
#include <stdio.h>
#include <math.h>
int main()
{
double x, y;
read:
     scanf("%lf",&x);
if (x < 0) goto read;
y = sqrt(x);
printf("%f\t""%f\n",x,y);
goto read;
}
```

Output:



```
5
5.000000         2.236068
4
4.000000         2.000000
3
3.000000         1.732051
```

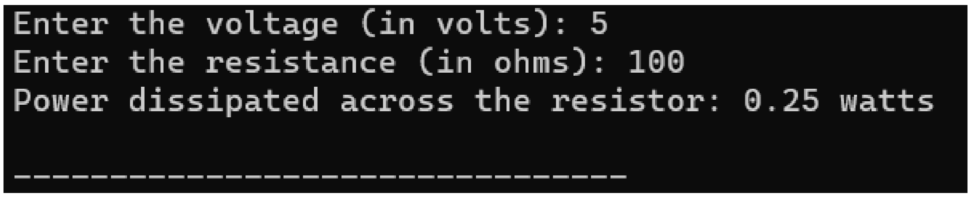
Task 1.16:

//while loop – power calculation using ohm’s law P=V^2/R

```
#include <stdio.h>
int main() {
    double voltage, resistance, power;
    printf("Enter the voltage (in volts): ");
    scanf("%lf", &voltage);
    // Ensure resistance is non-zero
    while (1) {/*this loop will indefintely continue until
```

```
     it encounters break statement*/
        printf("Enter the resistance (in ohms): ");
        scanf("%lf", &resistance);
        if (resistance > 0) {
            break; // Exit the loop if resistance is valid
        } else {
            printf("Resistance must be greater than 0.
Please enter a valid resistance.\n");
        }
    }
    power = (voltage * voltage) / resistance;
    printf("Power dissipated across the resistor: %.2lf
watts\n", power);
    return 0;
}
```

Output:



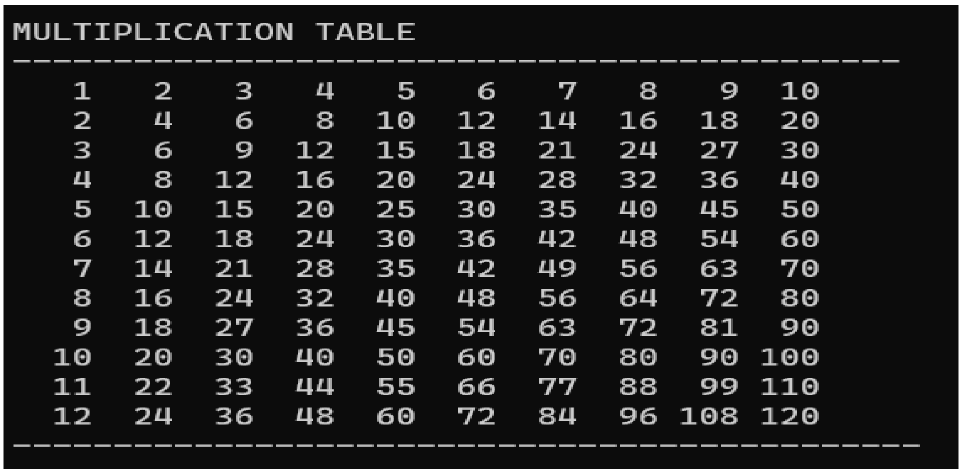
```
Enter the voltage (in volts): 5
Enter the resistance (in ohms): 100
Power dissipated across the resistor: 0.25 watts

--------------------------------
```

Task 1.17:

//do statement – multiplication table

```
#include <stdio.h>
#define COLMAX 10
#define ROWMAX 12
main()
{
int row,column, y;
row = 1;
printf("MULTIPLICATION TABLE \n");
printf("-------------------------------------------- \n");
do /*......OUTER LOOP BEGINS........*/
{
column = 1;
do /*.......INNER LOOP BEGINS.......*/
{
y = row * column;
printf("%4d", y);
column = column + 1;
}
while (column <= COLMAX); /*... INNER LOOP ENDS ...*/
printf("\n");
row = row + 1;
}
while (row <= ROWMAX);/*..... OUTER LOOP ENDS .....*/
printf("---------------------------------------------\n");
}
```

Output:

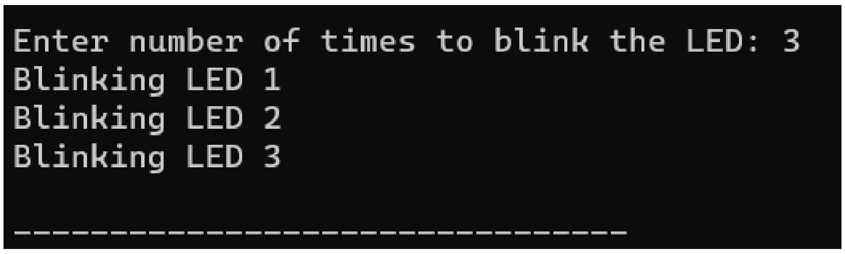


```
MULTIPLICATION TABLE
--------------------------------------------
   1   2   3   4   5   6   7   8   9  10
   2   4   6   8  10  12  14  16  18  20
   3   6   9  12  15  18  21  24  27  30
   4   8  12  16  20  24  28  32  36  40
   5  10  15  20  25  30  35  40  45  50
   6  12  18  24  30  36  42  48  54  60
   7  14  21  28  35  42  49  56  63  70
   8  16  24  32  40  48  56  64  72  80
   9  18  27  36  45  54  63  72  81  90
  10  20  30  40  50  60  70  80  90 100
  11  22  33  44  55  66  77  88  99 110
  12  24  36  48  60  72  84  96 108 120
---------------------------------------------
```

Task 1.18:

```
//for loop - Blinking a LED for certain number of times
#include <stdio.h>
int main() {
    int numBlinks;
     printf("Enter number of times to blink the LED: ");
     scanf("%d",&numBlinks);
     int i;
    // Simulate blinking an LED
    for (i = 0; i < numBlinks; ++i) {
        // In a real embedded system, this is where you would control the GPIO to toggle the LED
        // For simplicity, we'll just print a message to simulate the LED blinking
        printf("Blinking LED %d\n", i + 1);
    }
    return 0;
}
```

Output:



```
Enter number of times to blink the LED: 3
Blinking LED 1
Blinking LED 2
Blinking LED 3

--------------------------------
```

Take home tasks:

Task 1.19:

Create an authentication system which should ask the user to enter the password until the correct password is entered. (Hint: Use do, if-else, while)

Task 1.20:

Write a c code using ‘for’ loop and ‘if else’ condition to check the temperature from +10 degrees to -10 degrees. If the temperature is above 5, print the statement as high, if the temperature is below 0, print the statement as low and if the temperature is between 0 and 5, print the statement as normal.