

LAB EXERCISE 1: Install a C compiler on your system and configure the IDE. Write your first program to print "Hello, World!" and run it.

```
#include<stdio.h>
Int main()
{
    Printf("Hello World !");
Return 0;
}
```

Output : Hello World !

LAB EXERCISE 2: Write a C program that includes variables, constants, and comments. Declare and use different data types (int, char, float) and display their values.

```
//----- Single line comment
/*
    Multiline----- Comment
    #----- Pre processor
    Include----- Directive
    stdio.h----- Standard input output header file
    int-----Return type
    main-----Function name
    101-----Constants
*/
#include<stdio.h>//linking section
int main()
{

    int roll=100; //Declaration Of Variable
    long int fees=25000;
    float percentage = 78.90;
    char grade='C';

    printf("\nStudent's roll no = %d",roll);
    printf("\nStudent's fees = %ld",fees);
```

```
printf("\nStudent's Percentage = %.2f",percentage);
printf("\nStudent's grade = %c",grade);
return 0;
```

LAB EXERCISE 3: Write a C program that accepts two integers from the user and performs arithmetic, relational, and logical operations on them. Display the results.

```
#include <stdio.h>
int main() {
    int num1, num2;

    printf("Enter first integer: ");
    scanf("%d", &num1);
    printf("Enter second integer: ");
    scanf("%d", &num2);

    // Arithmetic Operations
    printf("\nArithmetic Operations:\n");
    printf("Addition (%d + %d) = %d\n", num1, num2, num1 + num2);
    printf("Subtraction (%d - %d) = %d\n", num1, num2, num1 - num2);
    printf("Multiplication (%d * %d) = %d\n", num1, num2, num1 * num2);
    if (num2 != 0) {
        printf("Division (%d / %d) = %d\n", num1, num2, num1 / num2);
        printf("Modulus (%d %% %d) = %d\n", num1, num2, num1 % num2);
    } else {
        printf("Division and Modulus: Cannot divide by zero\n");
    }

    // Relational Operations
    printf("\nRelational Operations:\n");
    printf("%d == %d is %d\n", num1, num2, num1 == num2);
    printf("%d != %d is %d\n", num1, num2, num1 != num2);
    printf("%d > %d is %d\n", num1, num2, num1 > num2);
    printf("%d < %d is %d\n", num1, num2, num1 < num2);
    printf("%d >= %d is %d\n", num1, num2, num1 >= num2);
    printf("%d <= %d is %d\n", num1, num2, num1 <= num2);
```

```

// Logical Operations
printf("\nLogical Operations:\n");
printf("(%d > 0) && (%d > 0) is %d\n", num1, num2, (num1 > 0) && (num2 > 0));
printf("(%d > 0) || (%d > 0) is %d\n", num1, num2, (num1 > 0) || (num2 > 0));
printf("!(%d > 0) is %d\n", num1, !(num1 > 0));

return 0;
}

```

-----Result-----

Enter first integer: 12

Enter second integer: 15

Arithmetic Operations:

Addition (12 + 15) = 27

Subtraction (12 - 15) = -3

Multiplication (12 * 15) = 180

Division (12 / 15) = 0

Modulus (12 % 15) = 12

Relational Operations:

12 == 15 is 0

12 != 15 is 1

12 > 15 is 0

12 < 15 is 1

12 >= 15 is 0

12 <= 15 is 1

Logical Operations:

(12 > 0) && (15 > 0) is 1

(12 > 0) || (15 > 0) is 1

!(12 > 0) is 0

LAB EXERCISE 4: Write a C program to check if a number is even or odd using an if-else statement. Extend the program using a switch statement to display the month name based on the user's input (1 for January, 2 for February, etc.).

```
#include <stdio.h>

int main() {
    int num;
    printf("Enter a number to check if it's even or odd: ");
    scanf("%d", &num);

    if (num % 2 == 0) {
        printf("%d is even.\n", num);
    } else {
        printf("%d is odd.\n", num);
    }
}

int month;
printf("\nEnter a number (1-12) to get the month name: ");
scanf("%d", &month);

switch (month) {
    case 1:
        printf("January\n");
        break;
    case 2:
        printf("February\n");
        break;
    case 3:
        printf("March\n");
        break;
    case 4:
        printf("April\n");
        break;
    case 5:
        printf("May\n");
        break;
    case 6:
        printf("June\n");
        break;
```

```

    case 7:
        printf("July\n");
        break;
    case 8:
        printf("August\n");
        break;
    case 9:
        printf("September\n");
        break;
    case 10:
        printf("October\n");
        break;
    case 11:
        printf("November\n");
        break;
    case 12:
        printf("December\n");
        break;
    default:
        printf("Invalid input! Please enter a number between 1 and 12.\n");
        break;
}

return 0;
}

```

-----Result -----

Enter a number to check if it's even or odd: 5
5 is odd.

Enter a number (1-12) to get the month name: 5
May

LAB EXERCISE 5: Write a C program to print numbers from 1 to 10 using all three types of loops (while, for, do-while).

```
#include <stdio.h>
int main() {
    // Using while loop
    printf("Using while loop:\n");
    int i = 1;
    while (i <= 10) {
        printf("%d ", i);
        i++;
    }
    printf("\n\n");

    // Using for loop
    printf("Using for loop:\n");
    for (int j = 1; j <= 10; j++) {
        printf("%d ", j);
    }
    printf("\n\n");

    // Using do-while loop
    printf("Using do-while loop:\n");
    int k = 1;
    do {
        printf("%d ", k);
        k++;
    } while (k <= 10);
    printf("\n");

    return 0;
}
```

-----Result-----

While Loop – 1 2 3 4 5 6 7 8 9 10

For Loop – 1 2 3 4 5 6 7 8 9 10

Do While Loop – 1 2 3 4 5 6 7 8 9 10

LAB EXERCISE 6: Write a C program that uses the break statement to stop printing numbers when it reaches 5. Modify the program to skip printing the number 3 using the continue statement.

```
#include <stdio.h>
int main() {
    int i;
    for (i = 1; i <= 10; i++) {
        if (i == 3) {
            continue;
        }
        if (i == 5) {
            break;
        }
        printf("%d Tops Technologies\n", i);
    }
    return 0;
}
```

LAB EXERCISE 7: Write a C program that calculates the factorial of a number using a function. Include function declaration, definition, and call.

```
#include <stdio.h>

/* Function declaration */
int fact(int n);

int main() {
    int num, result;

    printf("Enter the number = ");
    scanf("%d", &num);

    result = fact(num); /*Function call*/
    printf("Factorial of %d = %d", num, result);

    return 0;
}
```

```

/*Function definition */
int fact(int n) {
    int i, factorial = 1;
    for(i = 1; i <= n; i++) {
        factorial = factorial * i;
    }
    return factorial;
}

```

Output -----

Enter the number = 5

Factorial of 5 = 120

LAB EXERCISE 8: o Write a C program that stores 5 integers in a one-dimensional array and prints them. Extend this to handle a two-dimensional array (3x3 matrix) and calculate the sum of all elements.

```

#include<stdio.h>
int main()
{
    printf("One-dimensional array");
    int one_array[5]={1,2,3,4,5};

    printf("\n[1-5] Element");
    for(i=0;i<5;i++)
    {

        printf(" %d",one_array[i]);
    }
    printf("\n\n");
    int Two_array[3][3]={1,2,3},{4,5,6},{7,8,9}};
    int j,k;
    int sum=0;
    printf("Two-dimensional array (3x3 matrix)\n");
    for(j=0;j<3;j++)
    {
        for(k=0;k<3;k++)
        {
            printf("%d ",Two_array[j][k]);
sum += Two_array[j][k];
        }
        printf("\n");
    }
    printf("\n Sum of all Elements in matrix : %d",sum);
}

```



```
        return 0;
    }
```

Output-----

One-dimensional array
[1-5] Element 1 2 3 4 5

Two-dimensional array (3x3 matrix)
1 2 3
4 5 6
7 8 9

Sum of all Elements in matrix : 9

LAB EXERCISE 9 : Write a C program to demonstrate pointer usage. Use a pointer to modify the value of a variable and print the result.

```
#include <stdio.h>
int main()
{
    int num = 42;
    int *ptr = &num;

    printf("Original value: %d\n",num);
    printf("Address of number: %p\n", *ptr);

    *ptr = 46;
    printf("Modified value: %d\n",num);

    return 0;
}
```

Output-----

Original value: 42
Address of number: 000000000000002A
Modified value: 46

LAB EXERCISE 10: Write a C program that takes two strings from the user and concatenates them using strcat(). Display the concatenated string and its length using strlen().

```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[100], str2[100];
    printf("\nEnter the value str1 = ");
    gets(str1);

    printf("\nEnter the value str2 = ");
    gets(str2);

    printf("\nOriginal Value of str1 = %s", str1);
    printf("\nOriginal Value of str2 = %s", str2);

    strcat(str1, str2);

    printf("\nValue of str1 after using concat function = %s", str1);
    printf("\nValue of str2 after using concat function = %s", str2);
    printf("\nLength of str1 after concatenation = %lu", strlen(str1));

    return 0;
}
```

Output-----

Enter the value str1 = Ravi

Enter the value str2 = Rathod

Original Value of str1 = Ravi

Original Value of str2 = Rathod

Value of str1 after using concat function = RaviRathod

Value of str2 after using concat function = Rathod

Length of str1 after concatenation = 10

LAB EXERCISE 11: Write a C program that defines a structure to store a student's details (name, roll number, and marks). Use an array of structures to store details of 3 students and print them.

```
#include <stdio.h>
struct student {
    char name[100];
    int roll_number;
    int marks;
}s[100];

int main() {
    int i, count;

    printf("\nEnter the count of students = ");
    scanf("%d", &count);

    for(i = 0; i < count; i++) {
        printf("\nEnter the Student Name = ");
        scanf(" %[^\\n]", s[i].name);
        printf("Enter the Student Roll Number = ");
        scanf("%d", &s[i].roll_number);
        printf("Enter the Student Marks = ");
        scanf("%d", &s[i].marks);
    }

    printf("\n--- Student Details ---\\n");
    for(i = 0; i < count; i++) {
        printf("\\nName = %s", s[i].name);
        printf("\\nRoll Number = %d", s[i].roll_number);
        printf("\\nMarks = %d\\n", s[i].marks);
    }

    return 0;
}
```

Output-----

Enter the count of students = 2

Enter the Student Name = Ravi

Enter the Student Roll Number = 100

Enter the Student Marks = 77

Enter the Student Name = Rajveer

Enter the Student Roll Number = 101

Enter the Student Marks = 66

--- Student Details ---

Name = Ravi
Roll Number = 100
Marks = 77

Name = Rajveer
Roll Number = 101
Marks = 66

LAB EXERCISE 13 : Write a C program to create a file, write a string into it, close the file, then open the file again to read and display its contents.

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

```
int main()
```

```
{
```

```
    FILE *fp;
```

```
    fp = fopen("first.txt","w");
```

```
    fprintf(fp, "\nHello this is my second file");
```

```
    fclose(fp);
```

```
    return 0;
```

```
}
```

Display file -----

Hello this is my second file