**LAB EXERCISES**

1. **Real-World Applications of C Programming**
   * **Summary**: C is widely used in:
     1. **Embedded Systems**: For microcontrollers and hardware control.
     2. **Operating Systems**: UNIX, Linux, and other OSs are written in C.
     3. **Game Development**: Used in game engines for real-time performance and graphics.
2. **Install a C Compiler and Configure IDE**
   * **Summary**: Install GCC on your system and configure an IDE like VS Code, DevC++, or CodeBlocks. Write a basic "Hello, World!" program and run it.
3. **Write a C Program Using Variables, Constants, and Comments**
   * **Summary**: Write a C program that demonstrates the use of different data types (int, float, char), constants, and comments. Example program:

#include <stdio.h>

int main() {

int age = 30;

char initial = 'J';

float salary = 50000.75;

const double PI = 3.14159;

printf("Age: %d\n", age);

printf("Initial: %c\n", initial);

printf("Salary: %.2f\n", salary);

printf("PI: %.5f\n", PI);

return 0;

}

1. **Perform Arithmetic, Relational, and Logical Operations**
   * **Summary**: Write a program that takes two integers as input and performs arithmetic, relational, and logical operations on them. Example:

#include <stdio.h>

int main() {

int a, b;

printf("Enter two integers: ");

scanf("%d %d", &a, &b);

// Arithmetic Operations

printf("Sum: %d\n", a + b);

printf("Difference: %d\n", a - b);

// Relational Operations

printf("Are they equal? %d\n", a == b);

// Logical Operation

printf("Logical AND: %d\n", (a > 0) && (b > 0));

return 0;

}

**5. Control Flow Statements in C**

* **Lab Task**: Write a C program to:
  + **Check if a number is even or odd** using an if-else statement.
  + **Use a switch statement** to display the month name based on the user’s input (1 for January, 2 for February, etc.).

**Example**:

#include <stdio.h>

int main() {

int num, month;

// Check if number is even or odd

printf("Enter a number: ");

scanf("%d", &num);

if (num % 2 == 0) {

printf("Even\n");

} else {

printf("Odd\n"); }

// Use switch to display the month name

printf("Enter month number (1-12): ");

scanf("%d", &month);

switch(month) {

case 1: printf("January\n"); break;

case 2: printf("February\n"); break;

// other months

default: printf("Invalid month\n");

}

return 0;

}

**6. Looping in C**

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

**Example**:

#include <stdio.h>

int main() {

// Using while loop

int i = 1;

while (i <= 10) {

printf("%d ", i);

i++;

}

printf("\n");

// Using for loop

for (i = 1; i <= 10; i++) {

printf("%d ", i);

}

printf("\n");

// Using do-while loop

i = 1;

do {

printf("%d ", i);

i++;

} while (i <= 10);

return 0;

}

**7. Loop Control Statements in C**

* **Lab Task**: Write a C program that:
  + Uses the break statement to stop printing numbers when it reaches 5.
  + Modifies the program to **skip** printing the number 3 using the continue statement.

**Example**:

#include<stdio.h>

main(){

int i=1;

while(i<=5)

{  
printf("\n %d",i);

i++;  
if(i==4){

break;}

}

}

// Using continue to skip printing 3

#include<stdio.h>

main(){

int i=1;

while(i<=5)

{  
printf("\n %d",i);

i++;  
if(i==4){

continue;}

}

}

**8. What are functions in C? Explain function declaration, definition, and how to call a function. Provide examples.**

* In C programming, functions are blocks of code designed to perform a specific task. Functions help organize code, make it reusable, and improve readability.

1. Function Declaration

A function declaration tells the compiler what type of function it is and how it can be called, without needing to define the full function body. It’s also called a function prototype.

Syntax:-

return\_type function\_name(parameter\_list);

2. Function Definition

int add(int a, int b) { // Function definition

return a + b; }

3. Function Call

A function call is how you execute the function. You can call the function by using its name followed by arguments (if any).

Syntax:

function\_name(arguments);

**Example:-**

#include <stdio.h>  
// Function Declaration

int add(int, int);

main() {

int result = add(5, 10); // Function call

printf("The sum is: %d\n", result); // Output: The sum is: 15

}

// Function Definition

int add(int a, int b) {

return a + b; // Return the sum of a and b }

**9. Arrays in C**

* **Task**: Write a C program that:
  1. Stores 5 integers in a **one-dimensional array** and prints them.
  2. Extends this to handle a **two-dimensional array** (3x3 matrix) and calculates the sum of all elements.

**Example Code**:

#include<stdio.h>

main(){

int a[50],i,n;

printf("Enter size of array:");

scanf("%d",&n);

printf("Enter Elements:");

for(i=0;i<n;i++){

scanf("%d",&a[i]);

}

printf("\n Your Element are:");

for(i=0;i<n;i++){

printf("%d",a[i]);

}

}

**// Two-Dimensional Array: 3x3 matrix and calculating sum of elements**

#include<stdio.h>

main(){

int a[10][10],m,n,i,j,sum=0,sum2=0;

printf("Enter size of rows and colume");

scanf("%d %d",&m,&n);

printf("\n Enter Element...");

for(i=0;i<m;i++){

for(j=0;j<n;j++){

scanf("%d",&a[i][j]);

}

}

printf("\n Matrix is : \n ");

for(i=0;i<m;i++){

for(j=0;j<n;j++){

printf("%d ",a[i][j]);

}

printf(" \n ");

}

for(i=0;i<m;i++)

{

if(i % 2 == 0)

{ for(j=0;j<n;j++)

{ sum=sum+a[i][j];

}printf(" \n ");

}

}

printf("Sum of Even rows are: %d",sum);

for(i=0;i<m;i++){

for(j=0;j<n;j++){

if(j % 2 == 0)

sum2=sum2+a[i][j];

}

printf(" \n ");

} printf("Sum of Even colume are : %d",sum2);

}

**10. 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>

main(){

int a[5] = {1,2,3,4,5},\*p,i;

p=a;

for(i=0;i<5;i++)

{

printf("\n %d",\*p);

printf("\n %d Address is ",p);

p++; }

}

**11. 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>

main()

{

char str[50], str2[50];

int ans;

printf("Enter name first name: ");

gets(str);

printf("Enter name Last name: ");

gets(str2);

strcat(str,str2);

printf("Your full name is: ");

puts(str);

strlen(str);

puts(str);

}

**12. 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 firstName[50];

int roll;

float marks;

} s[3];

main() {

int i;

printf("Enter information of students:\n");

// storing information

for (i = 0; i < 3; ++i) {

s[i].roll = i + 1;

printf("\nFor roll number%d,\n", s[i].roll);

printf("Enter first name: ");

scanf("%s", s[i].firstName);

printf("Enter marks: ");

scanf("%f", &s[i].marks);

}

printf("Displaying Information:\n\n");

for (i = 0; i < 3; ++i) {

printf("\nRoll number: %d\n", i + 1);

printf("First name: ");

puts(s[i].firstName);

printf("Marks: %.1f", s[i].marks);

printf("\n");

}

return 0;

}

**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>

#include <stdlib.h>

main() {

FILE \*file;

char string[] = "This is the string.";

char buffer[100];

file = fopen("my\_file.txt", "w");

if (file == NULL) {

printf("Error opening file!\n");

return 1; // Indicate an error

}

fprintf(file, "%s", string);

fclose(file);

file = fopen("my\_file.txt", "r");

if (file == NULL) {

printf("Error opening file!\n");

return 1; // Indicate an error

}

printf("Contents of the file:\n");

while (fgets(buffer, sizeof(buffer), file) != NULL) {

printf("%s", buffer);

}

fclose(file);

}