**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>

int main() {

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

if (i == 5) {

break; // Exit the loop when i is 5

}

printf("%d ", i);

}

printf("\n");

// Using continue to skip printing 3

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

if (i == 3) {

continue; // Skip the number 3

}

printf("%d ", i); }

return 0;

}

**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>

int main() {

// One-Dimensional Array: Storing and printing 5 integers

int arr1D[5] = {10, 20, 30, 40, 50};

printf("One-Dimensional Array:\n");

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

printf("%d ", arr1D[i]);

}

printf("\n");

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

int arr2D[3][3] = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

printf("\nTwo-Dimensional Array (3x3 Matrix):\n");

int sum = 0;

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

for (int j = 0; j < 3; j++) {

printf("%d ", arr2D[i][j]);

sum += arr2D[i][j]; // Adding each element to the sum

}

printf("\n");

}

// Printing the sum of all elements

printf("\nSum of all elements in 3x3 matrix: %d\n", sum);

return 0;

}

**Explanation**:

* The program first defines a **one-dimensional array** (arr1D) with 5 integers, then it prints these values.
* It then defines a **two-dimensional array** (arr2D) as a 3x3 matrix and prints each element. It also calculates the sum of all elements by iterating through the matrix.