☑ Introduction to C

1. What is the use of C language in system-level programming?
2. List and explain at least 5 key features of C language.
3. Write a short note on the history and evolution of the C language.

☑ Environment Setup

1. How do you install and configure GCC compiler on Windows/Linux?
2. Write the steps to compile and run a C program using terminal.
3. List common installation/setup issues and their solutions.

☑ Your First C Program

1. Write a C program to print “Hello, World!”.
2. Modify your Hello World program to print your name and branch.
3. Write a program to display your college name and student ID.

☑ Data Types & Operators

1. Declare variables of type int, float, char, and double. Initialize them.
2. Write a program to demonstrate use of arithmetic and relational operators.
3. Show the difference between post-increment and pre-increment with an example.

☑ Variables & Constants

1. Demonstrate the use of #define and const to declare constants.
2. Write a program to explain scope of local and global variables.
3. Illustrate variable shadowing with a nested block.

☑ Control Flow (If-Else, Switch)

1. Write a program to check if a number is even or odd.
2. Write a program to find the largest of three numbers using if-else.
3. Use switch-case to implement a basic calculator (Add, Subtract, Multiply, Divide).

☑ Loops (for, while, do-while)

1. Write a for loop to print all odd numbers from 1 to 50.
2. Use while loop to reverse a given integer.
3. Use do-while loop to accept positive numbers until the user enters a negative number.

☑ Functions & Recursion

1. Write a function to calculate the factorial of a number.
2. Write a recursive function to print Fibonacci numbers.
3. Demonstrate call-by-value vs call-by-reference using swap function.

☑ Pointers & Memory Management

1. Declare a pointer and demonstrate pointer arithmetic.
2. Write a program to dynamically allocate memory for an integer array using malloc.
3. Demonstrate calloc and free in a small program.

☑ Structures & Unions

1. Define a struct student with fields name, roll, and marks. Take input and display it.
2. Demonstrate the difference between structure and union using memory usage.
3. Use typedef to simplify a structure definition.

☑ File Handling

1. Write a program to create a file and write your bio data into it.
2. Write a program to read from an existing file and print the content.
3. Handle a case where file does not exist (check using fopen).

☑ Bonus Practice Questions

1. Write a C program to toggle the nth bit of a number using bitwise operators.
2. Count the number of vowels in a string.

3. Create a menu-driven program to perform mathematical operations using switch-cas

Solutions

```
// ☑ Introduction to C
/*
1. What is the use of C language?
2. History of C and its features
3. Advantages of C
*/

// ☑ Environment Setup
/*
1. How to install GCC compiler on your system
2. Write, compile, and run a basic C program using command line
3. Common errors during setup and how to fix them
*/

// ☑ Your First C Program
#include <stdio.h>
int main() {
   printf("Hello, World!\n");
   return 0;
}

// ☑ Data Types & Operators
#include <stdio.h>
void data_types_and_operators() {
   int a = 10;
   float b = 5.5;
```

```
    char c = 'X';
    double d = 20.123;
    printf("Sum: %f\n", a + b);
    printf("Is a > b? %d\n", a > b);
    printf("Logical AND: %d\n", (a > 5 && b < 10));
}

// ☑ Variables & Constants
#include <stdio.h>
#define PI 3.14
void variables_constants() {
    const int x = 10;
    printf("PI: %f\n", PI);
    printf("Const x: %d\n", x);
    {
        int x = 20; // Shadowing
        printf("Shadowed x: %d\n", x);
    }
}

// ☑ Control Flow
#include <stdio.h>
void control_flow() {
    int n = 5;
    if (n % 2 == 0) printf("Even\n");
    else printf("Odd\n");

    int a = 10, b = 20, c = 15;
    if (a > b && a > c) printf("%d is largest\n", a);
    else if (b > c) printf("%d is largest\n", b);
    else printf("%d is largest\n", c);
```

```
    int choice = 2;
    switch(choice) {
        case 1: printf("Add\n"); break;
        case 2: printf("Subtract\n"); break;
        default: printf("Invalid\n");
    }
}

// ☑ Loops
#include <stdio.h>
void loops() {
    for (int i = 1; i <= 10; i++) printf("%d ", i);
    printf("\n");

    int num = 123, sum = 0;
    while (num != 0) {
        sum += num % 10;
        num /= 10;
    }
    printf("Sum of digits: %d\n", sum);

    int n = 5, i = 1;
    do {
        printf("%d x %d = %d\n", n, i, n*i);
        i++;
    } while (i <= 10);
}

// ☑ Functions & Recursion
#include <stdio.h>
```

```
int factorial(int n) {
    if (n == 0) return 1;
    return n * factorial(n - 1);
}
void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    printf("Inside swap: a=%d, b=%d\n", a, b);
}
void fibonacci(int n) {
    if (n <= 1) {
        printf("%d ", n);
        return;
    }
    int a = 0, b = 1, next;
    printf("0 1 ");
    for (int i = 2; i < n; i++) {
        next = a + b;
        printf("%d ", next);
        a = b;
        b = next;
    }
    printf("\n");
}

// ☑ Pointers & Memory Management
#include <stdio.h>
#include <stdlib.h>
void pointer_demo() {
    int x = 10;
```

```
    int *p = &x;
    printf("Value: %d, Address: %p\n", *p, p);
}
void pointer_arithmetic() {
    int arr[3] = {1, 2, 3};
    int *p = arr;
    for (int i = 0; i < 3; i++)
        printf("%d ", *(p + i));
    printf("\n");
}
void dynamic_memory() {
    int *ptr = (int*)malloc(5 * sizeof(int));
    for (int i = 0; i < 5; i++) ptr[i] = i * 2;
    for (int i = 0; i < 5; i++) printf("%d ", ptr[i]);
    free(ptr);
    printf("\n");
}

// ☑ Structures & Unions
#include <stdio.h>
#include <string.h>
struct Student {
    char name[20];
    int roll;
    float marks;
};
union Data {
    int i;
    float f;
};
typedef struct Student STU;
```

```
// ☑ File Handling
#include <stdio.h>
void file_write() {
   FILE *fp = fopen("data.txt", "w");
   fprintf(fp, "Hello File!\n");
   fclose(fp);
}
void file_read() {
   char ch;
   FILE *fp = fopen("data.txt", "r");
   if (!fp) {
      printf("File not found!\n");
      return;
   }
   while ((ch = fgetc(fp)) != EOF)
      putchar(ch);
   fclose(fp);
}

// ☑ Additional Practice Questions
#include <stdio.h>
void toggle_bit(int num, int pos) {
   int result = num ^ (1 << pos);
   printf("After toggling: %d\n", result);
}
void count_vowels(char *str) {
   int count = 0;
   for (int i = 0; str[i]; i++) {
      char ch = tolower(str[i]);
      if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') count++;
```

```
    }
    printf("Vowel count: %d\n", count);
}
void menu_program() {
    int choice, a = 5, b = 3;
    printf("1. Add\n2. Sub\n3. Mul\nEnter choice: ");
    scanf("%d", &choice);
    switch (choice) {
        case 1: printf("Sum = %d\n", a + b); break;
        case 2: printf("Sub = %d\n", a - b); break;
        case 3: printf("Mul = %d\n", a * b); break;
        default: printf("Invalid\n");
    }
}
```