Certainly! Let’s break down **Question 4: Logical Statements in C** in detail.

---

4. Logical Statements in C (Decision-Making Statements)

Logical statements (or decision-making statements) **control the flow** of execution in a program based on conditions.

There are three main types of **logical statements** in C:
1. **if statements**
2. **if-else statements**
3. **if-else-if ladder**
4. **Nested if statements**
5. **switch-case statement**

---

1. if Statement

The `if` statement checks a condition. If the condition is `true`, the block inside the `if` executes. Otherwise, it is skipped.

Syntax:

```
if (condition) {
    // Code executes if the condition is true
}
```

Example: if Statement

```
#include <stdio.h>
int main() {
    int num = 10;

    if (num > 0) {  // Condition: Is num greater than 0?
        printf("Number is positive\n");
    }

    return 0;
}
```

Output:

```
Number is positive
```

---

2. if-else Statement

The `if-else` statement provides an alternative block if the condition is `false`.

Syntax:

```
if (condition) {
    // Code executes if the condition is true
} else {
    // Code executes if the condition is false
}
```

Example: if-else Statement

```
#include <stdio.h>
int main() {
    int num = -5;

    if (num > 0) {
        printf("Number is positive\n");
    } else {
        printf("Number is negative\n");
    }

    return 0;
}
```

Output:

```
Number is negative
```

---

3. if-else-if Ladder

Used when there are **multiple conditions** to check in sequence.

Syntax:

```
if (condition1) {
    // Code executes if condition1 is true
} else if (condition2) {
    // Code executes if condition2 is true
} else {
    // Code executes if all conditions are false
}
```

Example: if-else-if Ladder

```
#include <stdio.h>
int main() {
    int marks = 75;

    if (marks >= 90) {
        printf("Grade: A\n");
    } else if (marks >= 75) {
        printf("Grade: B\n");
    } else if (marks >= 50) {
        printf("Grade: C\n");
    } else {
        printf("Grade: F\n");
    }

    return 0;
}
```

Output:

```
Grade: B
```

---

4. Nested if Statement

An `if` statement inside another `if` is called a **nested if statement**.

Syntax:

```
if (condition1) {
    if (condition2) {
        // Code executes if both condition1 and condition2 are true
    }
}
```

Example: Nested if Statement

```
#include <stdio.h>
int main() {
    int num = 10;

    if (num > 0) {
        if (num % 2 == 0) {
            printf("Number is positive and even\n");
        }
    }

    return 0;
}
```

Output:

```
Number is positive and even
```

---

5. switch-case Statement

The `switch` statement is used when a variable has **multiple possible values**. Instead of using multiple `if-else`, `switch` provides a **cleaner approach**.

Syntax:

```
switch (expression) {
    case value1:
        // Code executes if expression == value1
        break;
    case value2:
        // Code executes if expression == value2
        break;
    default:
        // Code executes if none of the cases match
}
```

Example: switch-case Statement

```
#include <stdio.h>
int main() {
    int day = 3;

    switch (day) {
        case 1:
            printf("Monday\n");
            break;
        case 2:
            printf("Tuesday\n");
            break;
        case 3:
            printf("Wednesday\n");
            break;
        case 4:
            printf("Thursday\n");
            break;
        default:
            printf("Invalid day\n");
    }

    return 0;
}
```

Output:

```
Wednesday
```

Explanation: Since `day = 3`, it matches `case 3`, so “Wednesday” is printed.

Key Differences Between if-else and switch-case

Feature	if-else	switch-case
Conditions	Can check relational (>, <, ==) and logical (&&, ||) conditions	Can only check equality (==)

Feature	if-else	switch-case
Data Types	Works with `int`, `float`, `char`, `string`, etc.	Works only with `int` and `char`
Execution	Checks all conditions sequentially	Jumps directly to the matched case
Performance	Slower for many conditions	Faster for multiple cases

---

Key Takeaways

✔ **if statement**: Executes if a condition is true.
✔ **if-else statement**: Executes one block if true, another if false.
✔ **if-else-if ladder**: Used when multiple conditions need checking.
✔ **Nested if**: if inside another if.
✔ **switch-case**: Efficient when checking a variable for multiple fixed values.

Would you like more examples or practice problems?