

1. Operator Precedence Demonstration:

```
#include <stdio.h>

int main() {
    int result = 5 + 10 * 2 - 8 / 4;
    printf("Result: %d\n", result);
    return 0;
}
```

Output : Result: 23

Sol: $5 + 10 * 2 - 8 / 4 = 5 + 20 - 2 = 23$

2. Parentheses Influence:

```
#include <stdio.h>

int main() {
    int result = (5 + 10) * 2 - 8 / 4;
    printf("Result: %d\n", result);
    return 0;
}
```

Output : Result: 28

Sol: $(5 + 10) * 2 - 8 / 4 = 15 * 2 - 2 = 30 - 2 = 28$

3. Bitwise Operator Precedence:

```
#include <stdio.h>

int main() {
    int x = 5, y = 3;
    int result = x & y | x << 1;
    printf("Result: %d\n", result);
    return 0;
}
```

Output : Result: 11

Sol: $x = 00000101$ (5) and $y = 00000011$ (3) so $x \& y = 00000001$ (1) now $x = 00000101$ (5) then shifting occurs here, $x \ll 1 = 00001010$ (10)
 $x \& y = 00000001$ (1) and $x \ll 1 = 00001010$ (10)
finally, $x \& y | x \ll 1 = 00001011$ (11)

4. Conditional Operator Exploration:

```
#include <stdio.h>

int main() {
    int a = 10, b = 15;
    int max = (a > b) ? a : b;
    printf("Maximum: %d\n", max);
    return 0;
}
```

Output : Maximum: 15

Sol: $\text{max} = (a > b)$ here is 15 and resultant is 15 now

5. Operator Precedence with Function Calls: (functions are advance topic, may discuss them in small)

```
#include <stdio.h>

int foo() {
    printf("Foo called\n");
    return 5;
}

int bar() {
    printf("Bar called\n");
    return 10;
}
```

Output : Foo called
Bar called
Result: 25

Sol: $\text{foo}() + \text{bar}() * 2 = 5 + 10 * 2 = 5 + 20 = 25$

```

}

int main() {
    int result = foo() + bar() * 2;
    printf("Result: %d\n", result);
    return 0;
}

```

6. Complex Expression Simplification:

```

#include <stdio.h>

int main() {
    int x = 5, y = 10, z = 15;
    int result = x + y * z / (x + y);
    printf("Result: %d\n", result);
    return 0;
}

```

Output : **Result: 15**

Sol: $x + y * z / (x + y) = 5 + 10 * 15 / (5 + 10) = 5 + 150 / 15 = 15$

7. Increment/Decrement Operators Effects:

```

#include <stdio.h>

int main() {
    int x = 5, y;
    y = x++ + x;
    printf("x: %d, y: %d\n", x, y);
    return 0;
}

```

Output : **x: 6, y: 11**

Sol: $y = x++ + x \rightarrow 6 + 5 = 11$, So $x = 6$ and $y = 11$

8. Logical Operators and Short-Circuiting:

```

#include <stdio.h>

int main() {
    int x = 5, y = 0;
    if (x && y++)
        printf("Inside if\n");
    printf("x: %d, y: %d\n", x, y);
    return 0;
}

```

Output : **x: 5, y: 1**

because: $y++$ is false (initially 0)

Sol: Since the if condition is false the code inside the if block (`printf("Inside if\n");`) is skipped the program directly executes the next statement (`printf("x: %d, y: %d\n", x, y);`)

9. Assignment Operators and Precedence:

```

#include <stdio.h>

int main() {
    int x = 5, y = 10;
    y = x += 3 * 2;
    printf("x: %d, y: %d\n", x, y);
    return 0;
}

```

Output : **x: 11, y: 11**

Sol: 1st $> 3 * 2 = 6$ 2nd $> x + = 6$ now becomes $x = x + 6$ finally $x = 11$ 3rd and finally $y = x$ so we get both x and y to be 11

10. Operator Precedence Quiz Game:

```

#include <stdio.h>

```

Output : **Result 1: 8**
Result 2: 2

Sol: Result 1: $5 + 6 / 2 = 5 + 3 = 8$
Result 2: $2 * 3 \% 4 = 6 \% 4 = 2$

```

int main() {
    int result1 = 5 + 6 / 2;
    int result2 = 2 * 3 % 4;
    printf("Result 1: %d\n", result1);
    printf("Result 2: %d\n", result2);
    return 0;
}

```

Operator Precedence and If-Else Tutorial

Program 1: Logical Operators and If-Else

```

#include <stdio.h>

int main() {
    int x = 5, y = 10;

    if (x > 3 || y > 15) {
        printf("Both conditions are true.\n");
    } else {
        printf("At least one condition is false.\n");
    }

    return 0;
}

```

Output: Both conditions are true.

Sol: We see $5 > 3$ equals true or 1 but $10 > 15$ is false or 0 so now || operator sees 1 or 0 which is 1.... we see the output of the first condition even though it is incorrect

Program 2: Ternary Operator and If-Else

```

#include <stdio.h>

int main() {
    int num = 7;

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

    return 0;
}

```

Output : 7 is odd.

Sol: clearly $\text{num} \% 2 \neq 0$ so 2nd Statement got executed

Output: 7 is odd.

Program 3: Compound Conditions and If-Else

```

#include <stdio.h>

int main() {
    int age = 18;

```

Output : The person is an adult male.

Sol: Here the 1st condition i.e. $\text{age} \geq 18 \ \&\& \ \text{gender} == 'M'$ is true because $\text{age} = 18$ and gender input value M satisfies the 1st statement, so 2nd condition does not meet

```

char gender = 'M';

if (age >= 18 && gender == 'M') {
    printf("The person is an adult male.\n");
} else {
    printf("The person is not an adult male.\n");
}

return 0;
}

```

Program 4: Nested If-Else

```

#include <stdio.h>

int main() {
    int x = 10, y = 5;

    if (x > y) {
        if (x % 2 == 0) {
            printf("x is even and greater than y.\n");
        } else {
            printf("x is odd and greater than y.\n");
        }
    } else {
        printf("x is not greater than y.\n");
    }

    return 0;
}

```

Output : x is even and greater than y.

Sol: int x = 10, y = 5 is the input data so loop checks x > y is true because 10 > 5 now it comes inside the loop and see the first condition that is x % 2 == 0 which is true since x = 10 so we get the desired output as "x is even and greater than y"