1. What will be the output?
```
int a = 5;
printf("%d", a++ + ++a);
```

A) 11
B) 12
C) 13
D) Undefined behavior

2. What is the output?
```
int x = 10;
x += x++ + ++x;
```

A) 31
B) 32
C) 33
D) Undefined behavior

3. What will be printed?
```
int a = 1, b = 2, c = 3;
printf("%d", a < b < c);
```

A) 0
B) 1
C) 2
D) Compilation error

4. What is the output?
```
int x = 5;
printf("%d", x << 1 + 1);
```

A) 10
B) 20
C) 40
D) 5

5. What will be the result?
```
int a = 8;
printf("%d", a >> 2);
```

A) 1
B) 2
C) 4
D) 8

6. What will be printed?
```
int a = 0;
printf("%d", a && a++);
```

A) 0
B) 1
C) Garbage value
D) Undefined behavior

7. What is the output?
```
int a = 5, b = 10;
printf("%d", a & b);
```

A) 0
B) 2
C) 5
D) 10

8. What will be printed?
```
int a = 4;
printf("%d", ~a);
```

A) 4
B) -4
C) -5
D) 5

9. What is the output?
```
int a = 3, b = 6;
printf("%d", a | b);
```

A) 3
B) 6
C) 7
D) 9

10. What will be printed?
```
int x = 2;
printf("%d", x *= 3 + 2);
```

A) 7
B) 10
C) 12
D) 14

11. What is the result?
```
int a = 5;
printf("%d", ++a * a++);
```

A) 30
B) 36
C) 42
D) Undefined behavior

12. What will be printed?
```
int a = 10;
printf("%d", a == 10 ? a++ : ++a);
```

A) 10
B) 11
C) 12
D) Undefined behavior

13. What is the output?
```
int x = 4;
printf("%d", x << 2);
```

A) 8
B) 12
C) 16

D) 32

14. What will be printed?
```
int a = 5, b = 3;
printf("%d", a ^ b);
```

A) 2
B) 4
C) 6
D) 8

15. What is the output?
```
int a = 1;
printf("%d", a++ + a++);
```

A) 2
B) 3
C) 4
D) Undefined behavior

16. What will be printed?
```
int a = 5;
printf("%d", a > 3 && a < 10);
```

A) 0
B) 1
C) 5
D) Compilation error

17. What is the result?
```
int a = 6;
printf("%d", a % 4);
```

A) 0
B) 1
C) 2
D) 3

18. What will be printed?
```
int a = 2;
printf("%d", a << a);
```

A) 4
B) 6
C) 8
D) 16

19. What is the output?
```
int a = 7;
printf("%d", a & 1);
```

A) 0
B) 1
C) 7
D) Undefined behavior

20. What will be printed?

```
int a = 0, b = 5;
printf("%d", a || b++);
```

A) 0
B) 1
C) 5
D) Undefined behavior

1. Answer: B
a++ gives 5, then a becomes 6. ++a makes it 7. So 5 + 7 = 12.

2. Answer: D
x is modified more than once in the same expression without a sequence point, so the behavior is undefined.

3. Answer: B
a < b gives 1, then 1 < c is true, so the result is 1.

4. Answer: B
1 + 1 is evaluated first. Then 5 << 2 = 20.

5. Answer: B
8 right-shifted by 2 bits gives 2.

6. Answer: A
Logical AND short-circuits. Since a is 0, a++ is not evaluated and the result is 0.

7. Answer: B
5 & 10 gives 2 in binary AND operation.

8. Answer: C
~4 gives -5 due to twoâ ™s complement representation.

9. Answer: C
3 | 6 gives 7 after bitwise OR.

10. Answer: B
3 + 2 is evaluated first. Then x = 2 * 5 = 10.

11. Answer: D
The variable a is modified twice in the same expression without a sequence point.

12. Answer: A
The condition is true, so a++ is used. It returns 10 and then increments.

13. Answer: C
4 << 2 shifts bits left by 2 positions, giving 16.

14. Answer: C
5 ^ 3 gives 6 using bitwise XOR.

15. Answer: D
The variable a is modified twice in the same expression, leading to undefined behavior.

16. Answer: B
Both conditions are true, so logical AND returns 1.

17. Answer: C
6 % 4 gives remainder 2.

18. Answer: C
2 << 2 equals 8.

19. Answer: B
7 & 1 checks the least significant bit. Since 7 is odd, the result is 1.

20. Answer: B
a is 0, so b++ is evaluated. Logical OR returns 1.