

## 20 Medium-to-Hard Pseudocode MCQs – Bitwise Operators

Assume 8-bit two's-complement integers unless otherwise stated.

Only ONE choice is guaranteed correct.

---

1.

```
a ← 0b10101100  
b ← 0b01010011  
print (a & b)
```

- A. 0b00000000
- B. 0b11111111
- C. 0b00000011
- D. 0b10101100

---

2.

```
x ← 0b11001100  
y ← x | 0b00110011  
print y
```

- A. 0b11111111
- B. 0b11001100
- C. 0b00110011
- D. 0b10000000

---

3.

```
m ← 0b11110000  
n ← m ^ 0b11110000  
print n
```

- A. 0b11110000
  - B. 0b00001111
  - C. 0b00000000
  - D. 0b11111111
-

4.

```
k ← 0b00001000  
print (k >> 2)
```

- A. 0b00000010
  - B. 0b00000011
  - C. 0b00001000
  - D. 0b00100000
- 

5.

```
p ← 0b00000011  
print (p << 4)
```

- A. 0b00110000
  - B. 0b00000011
  - C. 0b00001100
  - D. 0b11000000
- 

6.

```
a ← 0b10101010  
b ← 0b01010101  
c ← a ^ b  
print (c & 0x0F)
```

- A. 0b00001111
  - B. 0b00000000
  - C. 0b11110000
  - D. 0b11111111
- 

7.

```
x ← 0b11111111  
print (~x & 0xFF)
```

- A. 0b11111111
- B. 0b00000000

- C. 0b10000000
  - D. 0b01111111
- 

8.

```
val ← 0b00001111  
mask ← 0b11110000  
print ((val<< 4) | mask)
```

- A. 0b11111111
  - B. 0b00001111
  - C. 0b11110000
  - D. 0b00000000
- 

9.

```
a ← 0b11000011  
b ← 0b00111100  
print (a & b) | (a ^ b)
```

- A. 0b11111111
  - B. 0b11000011
  - C. 0b00111100
  - D. 0b00000000
- 

10.

```
x ← 0b10000000  
print (x >> 1)
```

- A. 0b01000000
  - B. 0b10000000
  - C. 0b11000000
  - D. 0b00000000
- 

11.

```
n ← 0b00001000  
count ← 0
```

```
while n  $\neq$  0 do
  n  $\leftarrow$  n & (n - 1)
  count  $\leftarrow$  count + 1
end while
print count
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
- 

12.

```
a  $\leftarrow$  0b10110010
b  $\leftarrow$  a & -a
print b
```

- A. 0b00000010
  - B. 0b00000001
  - C. 0b10110010
  - D. 0b01001101
- 

13.

```
x  $\leftarrow$  0b11100000
y  $\leftarrow$  (x >> 3) | (x << 5)
print (y & 0xFF)
```

- A. 0b00000111
  - B. 0b11111111
  - C. 0b00011111
  - D. 0b11100000
- 

14.

```
a  $\leftarrow$  0b01010101
b  $\leftarrow$  0b10101010
print (a ^ b) + 1
```

- A. 128
- B. 255

- C. 0
  - D. 256
- 

15.

```
x ← 0b11110000
y ← x & 0x0F
z ← y ^ 0xFF
print z
```

- A. 0b00001111
  - B. 0b11110000
  - C. 0b11111111
  - D. 0b00000000
- 

16.

```
a ← 0b01101001
b ← a | (a >> 1)
b ← b | (b >> 2)
b ← b | (b >> 4)
print (b & 0xFF)
```

- A. 0b01111111
  - B. 0b11111111
  - C. 0b00000001
  - D. 0b10000000
- 

17.

```
x ← 0b00000001
x ← x << 7
x ← x >> 7
print x
```

- A. 0b00000001
  - B. 0b10000000
  - C. 0b00000000
  - D. 0b11111111
-

18.

```
val ← 0b10101010  
parity ← 0  
while val ≠ 0 do  
    parity ← parity ^ 1  
    val ← val & (val - 1)  
end while  
print parity
```

A. 0

B. 1

C. 2

D. 4

---

19.

```
a ← 0b11001100  
b ← 0b00110011  
print ((a & b) << 1) + ((a | b) >> 1)
```

A. 0b11111111

B. 0b10011001

C. 0b01100110

D. 0b00000000

---

20.

```
x ← 0b00000000  
for i ← 0 to 7 do  
    x ← x | (1 << i)  
end for  
print x
```

A. 0b11111111

B. 0b00000000

C. 0b10000000

D. 0b01111111

---

## Answers

1 C

2 A

3 C

4 A

5 A

6 A

7 B

8 A

9 A

10 A

11 A

12 A

13 C

14 B

15 C

16 B

17 A

18 B

19 B

20 A