1. What does bit masking involve in C++?
   1. Performing arithmetic operations on bits
   2. Setting specific bits to 1 in a bit representation
   3. **Extracting, modifying, or setting individual bits**
   4. Shifting bits left and right
2. Which operator is used for bitwise AND in C++?
   1. **&**
   2. |
   3. ~
   4. ^
3. What is the result of 5 & 3 in binary?
   1. **1**
   2. 3
   3. 4
   4. 7
4. Which bitwise operator is used to toggle a bit from 0 to 1 and vice versa?
   1. &
   2. |
   3. **~**
   4. ^
5. What is the purpose of the bitwise NOT (~) operator?
   1. Bitwise AND
   2. Bitwise OR
   3. **Bitwise inversion**
   4. Bitwise XOR
6. What is the result of 12 | 9 in binary?
   1. **13**
   2. 8
   3. 12
   4. 9
7. How do you extract the least significant bit (LSB) of a number x?
   1. **x & 1**
   2. x | 1
   3. x >> 1
   4. x << 1
8. What is the result of 7 >> 2 in binary?
   1. 0
   2. **1**
   3. 2
   4. 3
9. How do you set the i-th bit of an integer x to 1?
   1. **x | (1 << i)**
   2. x & (1 << i)
   3. x ^ (1 << i)
   4. x >> i
10. Which operation clears a specific bit (sets it to 0) in an integer x?
    1. x | (1 << i)
    2. x & (1 << i)
    3. x ^ (1 << i)
    4. **x & ~(1 << i)**
11. Which operator is used to check if a specific bit is set in an integer x?
    1. ==
    2. !=
    3. **&**
    4. >>
12. How do you count the number of set bits (1s) in an integer x efficiently?
    1. Using a loop
    2. **Using the \_\_builtin\_popcount function (in GCC)**
    3. Using a recursive function
    4. Using the sizeof operator
13. What is the value of 1 << 5 in decimal?
    1. 1
    2. 16
    3. 5
    4. **32**
14. Which operation swaps the values of two variables a and b without using a temporary variable?
    1. **a = a ^ b; b = a ^ b; a = a ^ b;**
    2. a = a + b; b = a - b; a = a - b;
    3. a = a \* b; b = a / b; a = a / b;
    4. a = b; b = a;
15. What is the result of 15 ^ 7 in binary?
    1. **8**
    2. 6
    3. 0
    4. 1
16. How do you check if the rightmost (least significant) bit is set to 1 in an integer x?
    1. **(x & 1) == 1**
    2. (x & 1) == 0
    3. (x | 1) == 1
    4. (x ^ 1) == 1
17. Which operation clears the rightmost set bit (sets it to 0) in an integer x?
    1. **x & (x - 1)**
    2. x | (x - 1)
    3. x ^ (x - 1)
    4. x << 1
18. How do you check if an integer x is a power of 2?
    1. **(x & (x - 1)) == 0**
    2. (x & (x + 1)) == 0
    3. (x & (x >> 1)) == 0
    4. (x & (x << 1)) == 0
19. What is the binary representation of 10 in two's complement?
    1. **1111 0110**
    2. 1010 0101
    3. 0011 1010
    4. 1101 1011
20. Which bitwise operation is used to extract the leftmost (most significant) set bit in an integer x?
    1. **x & ~(x - 1)**
    2. x | (x - 1)
    3. x ^ (x - 1)
    4. x >> 1
21. What is the result of 12 >> 2 in binary?
    1. 0
    2. 1
    3. 2
    4. **3**
22. Which operator is used for bitwise OR in C++?
    1. &
    2. **|**
    3. ~
    4. ^
23. What is the result of 10 & 3 in binary?
    1. **2**
    2. 3
    3. 0
    4. 1
24. What is the result of 1 | 1 in binary?
    1. 6
    2. **1**
    3. 4
    4. 3
25. How do you check if the leftmost (most significant) bit is set to 1 in an integer x?
    1. **(x & (1 << (sizeof(int) \* 8 - 1))) == 1**
    2. (x & (1 << (sizeof(int) \* 8 - 1))) == 0
    3. (x | (1 << (sizeof(int) \* 8 - 1))) == 1
    4. (x ^ (1 << (sizeof(int) \* 8 - 1))) == 1
26. What is the result of 9 ^ 9 in binary?
    1. **0**
    2. 1
    3. 9
    4. 18
27. Which operation sets all bits in an integer x to 1?
    1. x & -1
    2. **x | -1**
    3. x ^ -1
    4. x >> -1
28. What is the result of 15 & 6 in binary?
    1. 15
    2. **6**
    3. 0
    4. 3
29. What is the return value of following function for arr[] = {9, 12, 2, 11, 2, 2, 10, 9, 12, 10, 9, 11, 2} and n is size of this array.

int fun(int arr[], int n)

{

int x = arr[0];

for (int i = 1; i < n; i++)

x = x ^ arr[i];

return x;

}

1. 0
2. **9**
3. 12
4. 2
5. Right shift(>>) and Left shift(<<) are equivalent to \_\_\_\_\_ by 2.
   1. Multiply and divide
   2. **Divide and multiply**
   3. Addition and subtraction
   4. Subtraction and addition