1.Write a C program to swap two numbers using bitwise XOR.

2.Implement a function to count the number of set bits (1s) in an integer.

3.Create a program to check if a given number is a power of two using bitwise operations.

4.Write a C function to toggle the nth bit of a given integer.

5.Implement a program to find the rightmost set bit in a binary representation of a number.

6.Create a function to check if two integers have opposite signs without using arithmetic operators.

7.Write a C program to find the bitwise AND of two integers without using the ‘&’ operator.

8.Implement a function to find the number of bits needed to represent an integer in binary.

9.Write a C function to rotate the bits of a given integer to the left by a specified number of positions.

10.Write a C program to efficiently multiply an integer by 7 without using the multiplication or addition operators, only bitwise operations.

11. Implement a C program to check if the kth bit is set or unset in a given integer.

12. Write a function to clear the rightmost set bit in a binary representation of an integer.

13. Create a program to count the number of bits required to convert integer A to integer B.

14. Implement a function to find the XOR of all numbers in a given range.

15. Write a C program to reverse the bits of a given integer.

16. Create a function to swap adjacent bits of a given integer.

17. Implement a program to find the parity (even or odd) of a binary number using bitwise operations.

18. Write a C function to check if a given number is a palindrome in binary representation.

19. Create a program to find the minimum XOR value between any two elements in an array.

20. Implement a function to set the rightmost unset bit in a binary representation of a number.