1. Produce in a program results of following operations
a) 1011 1001 & 1001 0110
b) 1101 1110 & 1100 0101
c) 0111 1101 | 1011 1110
d) 1100 0110 | 1101 1100
e) 1011 1001 ^ 1111 0110
f) 1100 0010 ^ 0000 0101
g) ~1011 1001 1001 0110
h) ~1101 1110 & 1100 0101
i) 1 << 7
j) pow(2, 7) >> 4

2. For the bit pattern given below for an integer, write code do the following opeations using bitwise operators.
1010 1110 1010 1011 0101 1101 0011 1011
a) Set 16th bit
b) Reset 28th bit
c) Read every odd bit and every even bit and copy these into continuous bits of another two separate variables and print their values.
d) Flip the 3rd nibble

3. Write a program that reads an integer and displays the number of bits set (i.e. the number of bits whose value is binary 1). Eg. If user enters 30, (binary 00000000 00011110) it should display 4 (because there are four binary 1s i.e. there are four bits 'set' in binary representation of decimal 30).

4. Using bitwise operators, write a program to store state of the following user settings of an OS.
Tablet Mode, WiFi, Mute, Airplane Mode, Auto Hide Taskbar.
All these settings have state which is either 'on' or 'off'.
The program should take user's input to
i) set value of these setting(s).
In this option, further input is taken from the user for the settings and stored. The user should be able to quit out of the option after setting zero or more settings.
ii) print status of settings.
In this option the program prints the current state of the user settings in a tabular format.
iii) exit the program
In this option the program exits.

5. On car company's website, the user is presented a page where various options are provided to choose the configuration of a car the user might be interested in. Once the user chooses various options, the data is sent to back-end of the company's website and the choices are stored.
Options shown to the user:
Interested in
[] Seat Covers
        []Beige
        []Dark
        []Dual Pattern
[] Alloys
[] Color

```
        []Coffee Brown
        []Pearl White
        []Marine Blue
        []Ash Grey
[] Steering Cover
[] Body Cover
```

Simulate this reading of user choices in client side such that minimum bits are used to send the choices to back-end.
Simulate back-end receiving the setting value by using a variable set to specific value. And extract the information present in the variable and show it in the console

```
Front End
---------
// capture the state of check boxes in a variable

// http.send(var);


Back End server
---------------
void Controller(char var)
{

}

Table in DB - Choices
-----------
UserID          SeatCovers | Beige | Dark | DualPatter | Alloys |
Color | Coffee Brown | ....
123             1                       1                 1        1
1


Common Understanding about bit pattern:
[] Seat Covers -> LSB - 0
        []Beige -
        []Dark  -
        []Dual Pattern - 1,2 => seat cover choice 00 -> Beige, 01-
>Dark, 10->Dual
[] Alloys => 3
[] Color => 4
        []Coffee Brown -
        []Pearl White -
        []Marine Blue -
        []Ash Grey -    5,6 => color choice, 00 -> CB, 01 -> PW, 10 -
>MB 11->AG
[] Steering Cover - 7

void FrontEnd()
{
        char input{};

        // assume following choices by user:
```

```
// Seat Cover -> Dark
// Alloy -> not set
// Color -> Marine Blue
// Steering Cover -> not set

char bcv {};
if (/*seat cover option is checked by user*/)
{
        bcv = 1;        // 0000 0001
        input |= bcv;
        // set the choice of seat cover
        if (beige == checked)
        {
                bcv = 0;
                input |= bcv;
        }
        else if (dark == checked)
        {
                bcv = 1;        // 0000 0001
                bcv <<= 1;      // 0000 0010
                input |= bcv;
        }
        else if (dual == checked)
        {
                bcv = 1;        // 0000 0001 -> 0000 0100
                bcv << 2;       // 0000 0100
                input |= bcv;
        }
}
if (alloy == checked)
{
        bcv = 1;
        bcv <<= 3;
        input |= 2;
}
if (color == checked)
{
        bcv = 1;
        bcv <<= 4;
        input |= bcv;
        if (cb == checked)
        {
                // no code required to set bit 5,6 to 00
        }
        else if (pw == checked)
        {
                bcv = 1; // 0000 0001 -> 0010 0000
                bcv <<= 5;
                input |= bcv;
        }
        else if (mb == checked)
        {
                bcv = 1; // 0000 0001 -> 0100 0000
                bcv <<= 6;
                input |= bcv;
        }
        else if (ag == checked)
```

```
                {
                        bcv = 0b11; // 0000 0011 -> 0110 0000
                        bcv <<= 7;
                        input |= bcv;
                }
        }
        if (sc == checked)
        {
                bcv = 1; // 0000 0001 -> 1000 0000
                bcv <= 8;
                input |= bcv;
        }

// input variable now has all the users choices captured
// send input to backend
}

void BackEnd(char input)
{
        char bcv{1}; // 0000 0001
        // check for seat cover
        if (input & bcv)
        {
                // set seat cover column to 1
                // check for which seat cover?
                bcv = 0b10;     // 0000 0010 & 0101 0011
                if (input & bcv)
                {
                        // set seat cover choice to dark
                }
                else
                {
                        // bit with index 1 is not set
                        bcv = 1; //0000 0001
                        bcv <<= 2; // 0000 0100 & 0101 0101
                        if (input & bcv)
                        {
                                // set seat cover choice for dual to 1
                        }
                        else
                        {
                                // set seat cover choice for beige to
1
                        }
                }
                // ALTERNATIVE
                bcv = 0b01;
                bcv << 1; // 0000 0010
                auto i = input & bcv;   // 0101 0010 & 0000 0010 =>
0000 0010 >> 1 => 0000 0001
                i >>= 1;
                if (i == 1) => set seat cover choice to dark
                bcv = 0b01;
                bcv << 2; // 0000 0100
                i = input & bcv;        // 0101 0100 & 0000 0100 =>
0000 0100 >> 2 => 0000 0001
                if (i == 1) => set seat cover choice to dual
```

```
                // 0000 0110
                bcv = 0b11;
                bcv <<= 1; // 0000 0110
                auto i = input & bcv; 0101 0110 & 0000 0110 => 0000
0110
                i >>= 1; 0000 0011
                if (i == 3) => choice of next type of seat cover
        }
        else
                // set seat cover, and seat cover choice fields in DB
to 0


}
```