1 Objective(s)

- To convert decimal (dotted decimal ip) to binary (32 bit ip) and vice-versa manually by creating a user-defined function.

2 Problem analysis

2.1 Decimal to Binary

To convert numbers from decimal to binary, the given decimal number is divided repeatedly by 2 and the remainders are noted down till we get 0 as the final quotient. The following steps is considered as the decimal to binary formula that shows the procedure of conversion.
Step 1: Divide the given decimal number by 2 and note down the remainder.
Step 2: Now, divide the obtained quotient by 2, and note the remainder again.
Step 3: Repeat the above steps until you get 0 as the quotient.
Step 4: Now, write the remainders in such a way that the last remainder is written first, followed by the rest in the reverse order.
Step 5: This can also be understood in another way which states that the Least Significant Bit (LSB) of the binary number is at the top and the Most Significant Bit (MSB) is at the bottom. This number is the binary value of the given decimal number.
Let us understand this with an example.
Example: Convert the decimal number 1310 to binary.
Solution: We will start dividing the given number (13) repeatedly by 2 until we get the quotient as 0. We will note the remainders in order.

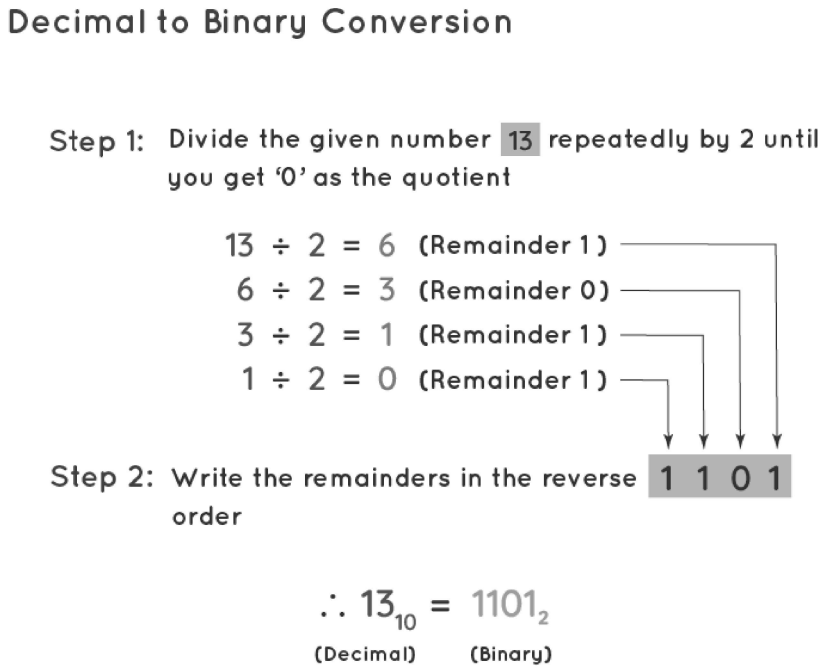

Figure 1: Decimal to Binary Conversion

2.2 Binary to Decimal

The positional notation method is one in which the value of a digit in a number is determined by a weight based on its position. This is achieved by multiplying each digit by the base(2) raised to the respective power depending upon the position of that digit in the number. The summation of all these values obtained for each digit gives the equivalent value of the given binary number in the decimal system.

Observe the following steps to understand the binary to decimal conversion. Let us consider the binary number (101101)2. In any binary number, the rightmost digit is called the 'Least Significant Bit' (LSB) and the left-most digit is called the 'Most Significant Bit' (MSB). For a binary number with 'n' digits, the least significant bit has a weight of 20 and the most significant bit has a weight of 2n-1.

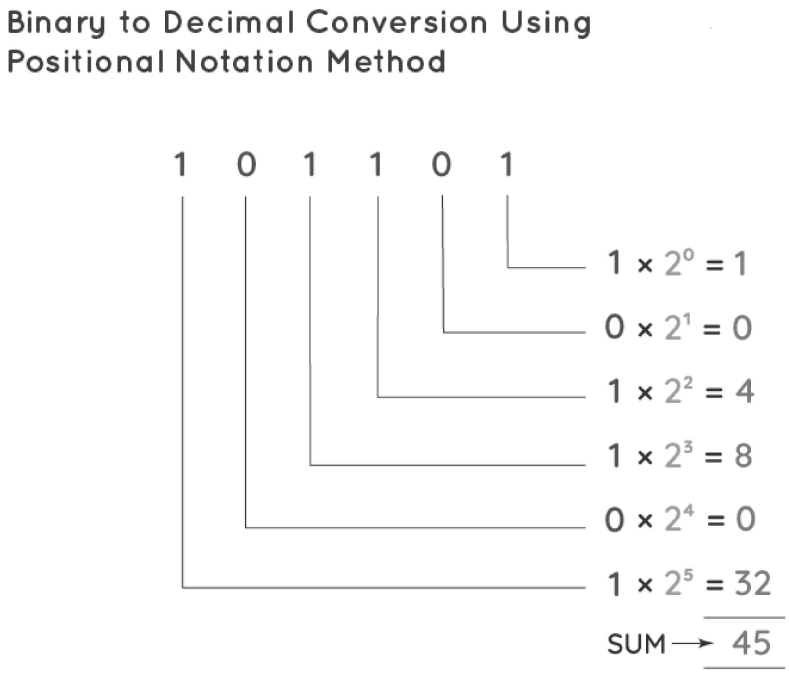

Figure 2: Binary to Decimal Conversion

3 Code Implementation in C (Decimal to Binary)

```c
#include <stdio.h>

// Function to convert decimal to binary
void decimalToBinary(int num) {
    // Array to store binary number
    int binaryNum[32];

    // Counter for binary array
    int i = 0;
    while (num > 0) {
        // Store the remainder in binary array
        binaryNum[i] = num % 2;
        num = num / 2;
        i++;
    }

    // Print binary array in reverse order
    printf("Binary representation: ");
    for (int j = i - 1; j >= 0; j--)
        printf("%d", binaryNum[j]);
    printf("\n");
}

int main() {
```

```
    int decimalNum;

    // Input the decimal number
    printf("Enter a decimal number: ");
    scanf("%d", &decimalNum);

    // Call function to convert decimal to binary
    decimalToBinary(decimalNum);

    return 0;
}
```

4 Sample Input/Output (Compilation, Debugging & Testing)

Enter a decimal number: 13
13 in binary: 1101

5 Code Implementation in C (IPv4 address Decimal to Binary)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Function to convert IPv4 address to binary
void ipv4_to_binary(char *ipv4_address) {
    char *token;
    int binary[32] = {0};
    int i = 0;

    // Tokenize the IPv4 address
    token = strtok(ipv4_address, ".");
    while (token != NULL) {
        int octet = atoi(token); // Convert token to integer
        // Convert octet to binary
        for (int j = 7; j >= 0; j--) {
            binary[i++] = (octet >> j) & 1;
        }
        token = strtok(NULL, ".");
    }

    // Print the binary representation
    for (int k = 0; k < 32; k++) {
        printf("%d", binary[k]);
        if ((k + 1) % 8 == 0) {
            printf(" ");
        }
    }
    printf("\n");
}

int main() {
    char ipv4_address[16]; // IPv4 address string (e.g., "192.168.1.1")

    printf("Enter an IPv4 address: ");
    fgets(ipv4_address, sizeof(ipv4_address), stdin);
```

```
37      ipv4_address[strcspn(ipv4_address, "\n")] = '\0'; // Remove newline
            character if present
38
39      printf("Binary representation: ");
40      ipv4_to_binary(ipv4_address);
41
42      return 0;
43  }
```

6 Sample Input/Output (Compilation, Debugging & Testing)

Enter an IPv4 address: 192.168.1.1
Binary Representation: 11000000.10101000.00000001.00000001

7 Discussion & Conclusion

Based on the focused objective(s) to understand about the algorithms, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

8 Lab Task (Please implement yourself and show the output to the instructor)

1. Convert an ipv4 dotted binary ip to 32-bit decimal ip.

9 Lab Exercise (Submit as a report)

- Write a C program that prompts the user to enter an IP address belonging to a specific class (A, B, or C) and then offers options to convert it into binary or vice versa

10 References

1. https://www.cuemath.com/numbers/decimal-to-binary/
2. https://www.cuemath.com/numbers/binary-to-decimal/
3. https://www.programiz.com/c-programming/examples/binary-decimal-convert

11 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.