

Task-10

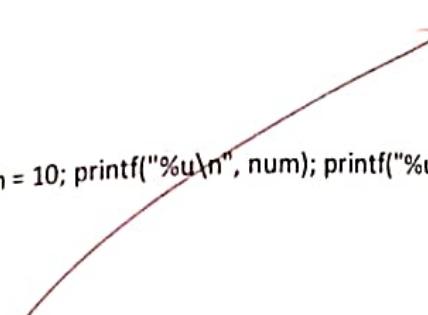
Bit Map - Introduction, XOR, AND, OR, right shift, left shift.

Algorithm

1. Take the 32-bit unsigned integer as input in binary format.
2. Convert the binary string into a character array.
3. Initialize two pointers; one pointing to the beginning of the array and another pointing to the end of the array.
4. Swap the bits at the two pointers and increment the first pointer and decrement the second pointer until they meet or cross each other.
5. Convert the modified character array back to a binary string.
6. Convert the binary string to an integer and return the result.

```
#include <stdio.h>
unsigned int reverseBits(unsigned int num) { unsigned int reversed = 0;
    int bits = sizeof(num) * 8;
    for(int i = 0; i < bits; i++) { if (num & (1 << i)) {
        reversed |= 1 << (bits - 1 - i);
    }
}
return reversed;
}

int main() {
    unsigned int num = 10; printf("%u\n", num); printf("%u\n", reverseBits(num)); return 0;
}
```



## Task-1DB

To write a function that takes an unsigned integer and returns the number of 1 bits.

Algorithm for counting numbers of 1 bits

Algorithm for Variable Count to 0.

1. Initialize a variable Count to 0.

2. Loop through each bit in the 32-bit integer.

3. If the current bit is a 1, increment the count by 1.

4. After looping through all the bits, return the count.

Algorithm for computing the Hamming distance between two integers:

1. Initialize a Variable Count to 0.

2. Loop through each bit in the 32-bit integers x and y.

3. If the current bit in x is different from the current bit in y, increment the count by 1.

4. After looping through all the bits, return the count.

```

#include <stdio.h>
unsigned int hammingDistance(unsigned int x, unsigned int y)

{
    unsigned int dist = 0;
    unsigned int val = x ^ y; // bitwise XOR of x and y

    while (val) {
        // count the number of set bits (i.e., 1s) in val
        dist++;
        val &= val - 1;
    }

    return dist;
}

int main() {
    unsigned int x = 10; unsigned int y = 15;
    printf("%u\n", hammingDistance(x, y)); return 0;
}

```

VEL TECH - CSE	
S.NO	D
PERFORMANCE (5)	8
RESULT AND ANALYSIS (3)	2
VIVA VOCE (3)	2
REPORT (4)	4
TOTAL (15)	14
SIGN WITH DATE	

✓ D ✓

Result:

Thus, the program is executed and verified successfully.