

Task 10: Bit manipulation - Introduction, XOR, AND, OR right shift, left shift

Reverse bits of a given 32-bit unsigned integer.

example:

Input: $n = 11111111111111111111111111111101$

Output: 964176192 (001100101110000010100101000)

Explanation: The input binary string 000000101001010001011100100111 represents the unshifted.

001110010111100000101000101000000.

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 other pointing to the end of the array.
4. Convert the modified character array back to the binary string.
5. Convert the binary string to an integer and return the result.

Program: ~~Now how between 01 performing bit shift~~ ~~11001100110011001100110011001100~~

#include <cs50.h>

unsigned int reversebits(unsigned int num)

{ unsigned int reversed = 0;

int bits = size of (num) ;

```

int i=0; i<6bit; i++)
{
    if (num & (1<<i))
        reversed = 1<<(6bit-i-1);
    return reversed;
}

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

```

10
1342177280
write a function that takes an unsigned integer and returns the no. of 1's bits has (also known as the Hamming weight).

example:

Input: n=1, j=4

Output: 2

Explanation:

1(0001)

4(0100)



The above arrows point to positions where the corresponding bits are different.

Algorithm

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 count by 1.
4. After looping through all the bits, return the count.

Program:

```
#include <stdio.h>
unsigned int Hamming distance(unsigned int x, unsigned int y)
{
    unsigned int dist = 0;
    unsigned int val = x ^ y;
    while (val) {
        dist++;
        val = val & (val - 1);
    }
    return dist;
}

int main()
{
    unsigned int x = 10;
    unsigned int y = 15;
    printf("The Hamming distance between %d and %d is %d\n", x, y, Hamming distance(x, y));
    return 0;
}
```

Output

?

Result Thus the program is executed and verified successfully.

VEL TECH - CSE	
EX NO.	102
PERFORMANCE (5)	8
RESULT AND ANALYSIS (3)	9
VIVA VOCE (3)	9
RECORD (4)	4
TOTAL (15)	30
S. N WITH DATE	2/1/2022