

Module: R1: C Programming**Section: C Memory Management & Usage Task: Bitwise Operation****Task 5.1****Bitwise Operation****1. Explanation:**

1. The program prompts the user to enter a number.
2. It initializes a counter to keep track of the number of trailing zeros.
3. Using bitwise AND (&) operation, the program checks the least significant bit (LSB) of the number (i.e., the rightmost bit) by performing $(n \& 1)$.
4. If the result of $(n \& 1)$ is equal to zero (meaning the LSB is zero), the program increments the counter.
5. The program continues shifting the number to the right by one bit $(n \gg= 1)$ until it encounters the first occurrence of the bit '1'.
6. The final count represents the total number of trailing zeros in the given number.

2. Code Snippets:

```

/*
 * Author: Noman Rafiq
 * Dated: June 26, 2024
 * Description: The programs prompts the user to Enter a number.
Then the program uses a counter to count the number of zeroes in the
right most position of the number using bitwise & operation.
    The counter keeps incrementing by 1 until the first occurrence of
the bit '1' is reached in the number.
*/

#include <stdio.h>
int trailing_zeros(int n){
    int count=0;

    //Keep Repeating until we find first '1' in the right most position
of the number
    while ((n & 1) == 0){
        count++;
        n >>= 1;
    }
    return count;
}

```

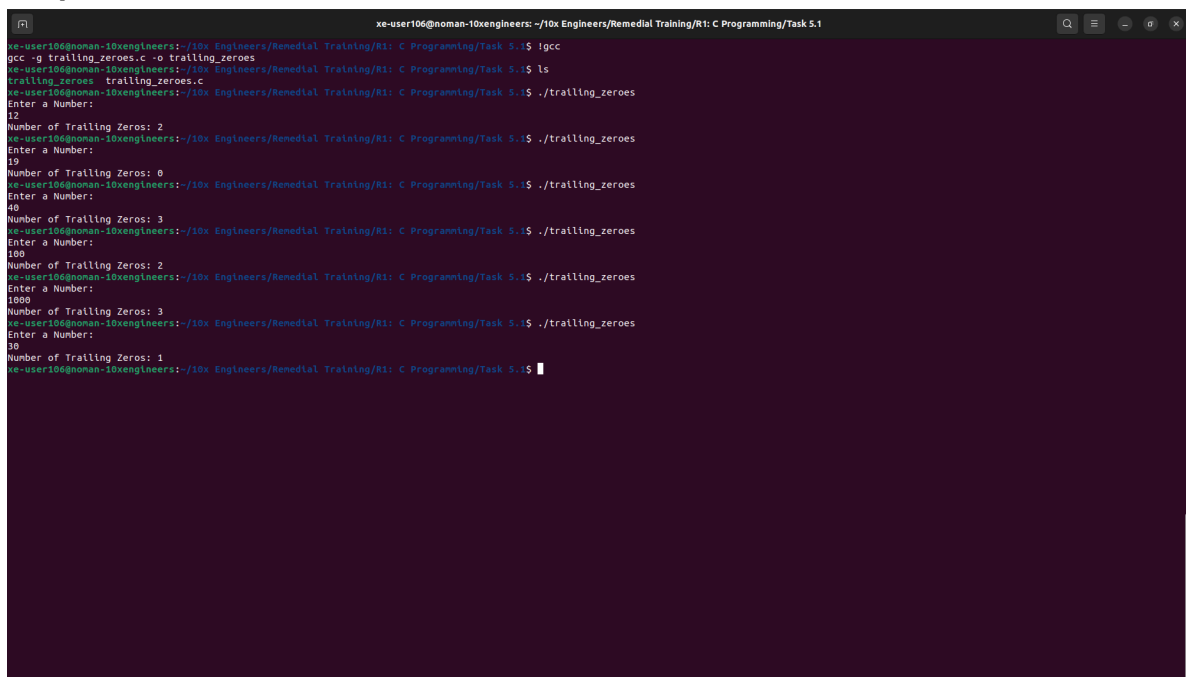
```
int main() {
    unsigned int x;
    printf("Enter a Number: \n");
    scanf("%u", &x);

    //Trailing Zero's
    int n = trailing_zeros(x);

    printf("Number of Trailing Zeros: %d\n", n);

    return 0;
}
```

3. Output:



```
xe-user106@noman-10xengineers: ~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ gcc
gcc -g trailing_zeros.c -o trailing_zeros
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ls
trailing_zeros  trailing_zeros.c
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ./trailing_zeros
Enter a Number:
12
Number of Trailing Zeros: 2
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ./trailing_zeros
Enter a Number:
19
Number of Trailing Zeros: 0
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ./trailing_zeros
Enter a Number:
48
Number of Trailing Zeros: 3
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ./trailing_zeros
Enter a Number:
100
Number of Trailing Zeros: 2
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ./trailing_zeros
Enter a Number:
1000
Number of Trailing Zeros: 3
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$ ./trailing_zeros
Enter a Number:
16
Number of Trailing Zeros: 1
xe-user106@noman-10xengineers:~/10x Engineers/Remedial Training/R1: C Programming/Task 5.1$
```