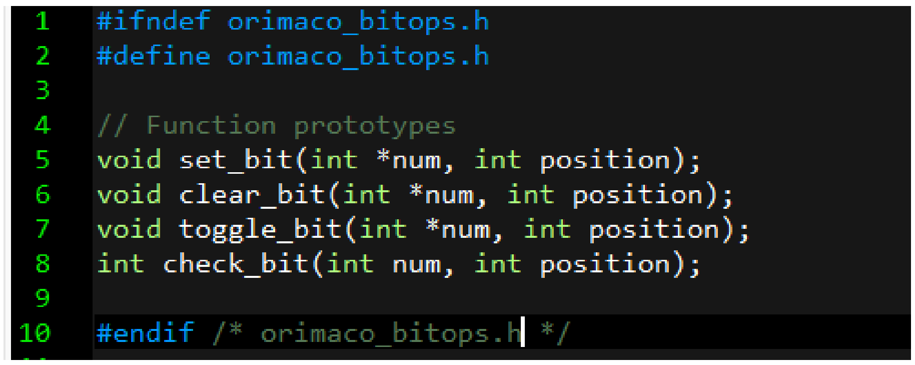
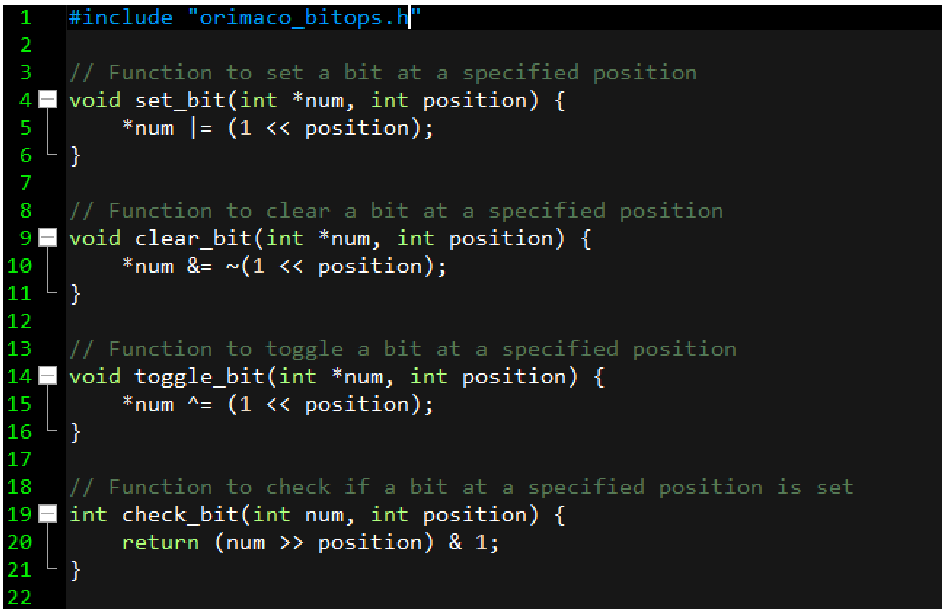


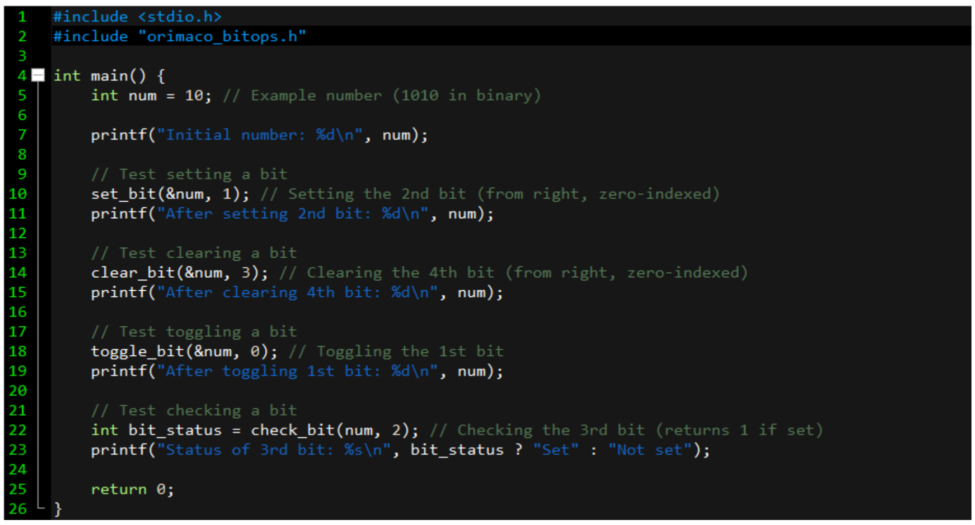
```c
#ifndef orimaco_bitops.h
#define orimaco_bitops.h

// Function prototypes
void set_bit(int *num, int position);
void clear_bit(int *num, int position);
void toggle_bit(int *num, int position);
int check_bit(int num, int position);

#endif /* orimaco_bitops.h */
```

Header File



```c
#include "orimaco_bitops.h"

// Function to set a bit at a specified position
void set_bit(int *num, int position) {
    *num |= (1 << position);
}

// Function to clear a bit at a specified position
void clear_bit(int *num, int position) {
    *num &= ~(1 << position);
}

// Function to toggle a bit at a specified position
void toggle_bit(int *num, int position) {
    *num ^= (1 << position);
}

// Function to check if a bit at a specified position is set
int check_bit(int num, int position) {
    return (num >> position) & 1;
}

```

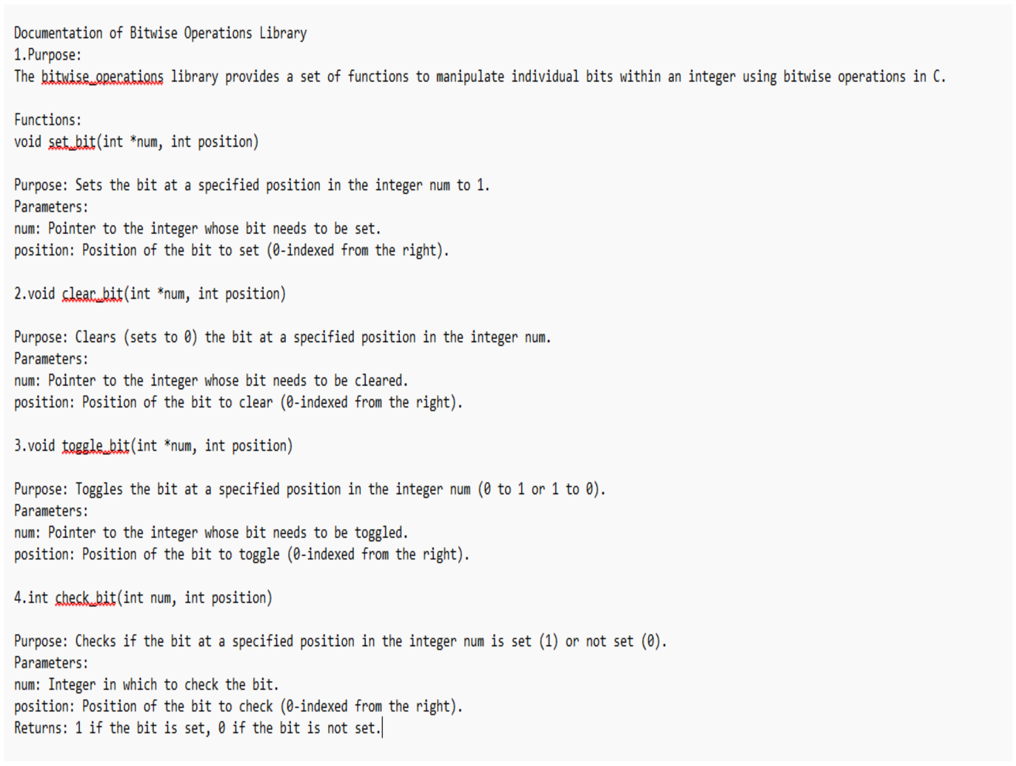
Implementation File



```c
#include <stdio.h>
#include "orimaco_bitops.h"

int main() {
    int num = 10; // Example number (1010 in binary)

    printf("Initial number: %d\n", num);

    // Test setting a bit
    set_bit(&num, 1); // Setting the 2nd bit (from right, zero-indexed)
    printf("After setting 2nd bit: %d\n", num);

    // Test clearing a bit
    clear_bit(&num, 3); // Clearing the 4th bit (from right, zero-indexed)
    printf("After clearing 4th bit: %d\n", num);

    // Test toggling a bit
    toggle_bit(&num, 0); // Toggling the 1st bit
    printf("After toggling 1st bit: %d\n", num);

    // Test checking a bit
    int bit_status = check_bit(num, 2); // Checking the 3rd bit (returns 1 if set)
    printf("Status of 3rd bit: %s\n", bit_status ? "Set" : "Not set");

    return 0;
}
```

Test Program

Documentation



Documentation of Bitwise Operations Library
1.Purpose:
The bitwise_operations library provides a set of functions to manipulate individual bits within an integer using bitwise operations in C.

Functions:
void set_bit(int *num, int position)

Purpose: Sets the bit at a specified position in the integer num to 1.
Parameters:
num: Pointer to the integer whose bit needs to be set.
position: Position of the bit to set (0-indexed from the right).

2.void clear_bit(int *num, int position)

Purpose: Clears (sets to 0) the bit at a specified position in the integer num.
Parameters:
num: Pointer to the integer whose bit needs to be cleared.
position: Position of the bit to clear (0-indexed from the right).

3.void toggle_bit(int *num, int position)

Purpose: Toggles the bit at a specified position in the integer num (0 to 1 or 1 to 0).
Parameters:
num: Pointer to the integer whose bit needs to be toggled.
position: Position of the bit to toggle (0-indexed from the right).

4.int check_bit(int num, int position)

Purpose: Checks if the bit at a specified position in the integer num is set (1) or not set (0).
Parameters:
num: Integer in which to check the bit.
position: Position of the bit to check (0-indexed from the right).
Returns: 1 if the bit is set, 0 if the bit is not set.