Bit Operations

In the following questions, use bit manipulation operations to achieve the intended functionality and fill out the function details -

(a) Implement a function `isPalindrome` which checks if the binary representation of a given number is palindrome. The function returns true if and only if the binary representation of `num` is a palindrome. Assume `num` is 32 bits.

For example, the function should return true for `isPalindrome(0xDEADDAED)` since binary representation of 9 is `1001` which is a palindrome.

```
/**
* Returns true if binary representation of num is a palindrome
*/
public static boolean isPalindrome(int num) {

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________
}
```

(b) Implement a function swap which for a given integer, swaps two bits at given positions. The function returns the resulting integer after bit swap operation.

For example, when the function is called with inputs swap(31, 3, 7), it should reverse the 3rd and 7th bits from the right and return 91 since 31 (00011111) would become 91 (01011011).

```
/**
* Function to swap bits at position a and b (from right) in integer num
*/
public static int swap(int num, int a, int b) {
    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    _________________________________________________

    return num;
}
```