```c
#include <stdio.h>

int main(void)
{
    //function prototypes
    void PrintBinaryFormOfNumber(unsigned int);

    //variable declarations
    unsigned int a;
    unsigned int num_bits;
    unsigned int result;

    //code
    printf("\n\n");
    printf("Enter An Integer = ");
    scanf("%u", &a);

    printf("\n\n");
    printf("By How Many Bits Do You Want To Shift A = %d To The Left ? ", a);
    scanf("%u", &num_bits);

    printf("\n\n\n\n");
    result = a << num_bits;
    printf("Bitwise LEFT-SHIFT By %d Bits Of A = %d \nGives The Result = %d (Decimal).\n\n", num_bits, a, result);
    PrintBinaryFormOfNumber(a);
    PrintBinaryFormOfNumber(result);

    return(0);
}

// ****** BEGINNERS TO C PROGRAMMING LANGUAGE : PLEASE IGNORE THE CODE OF THE FOLLOWING FUNCTION SNIPPET 'PrintBinaryFormOfNumber()' ******
// ****** YOU MAY COME BACK TO THIS CODE AND WILL UNDERSTAND IT MUCH BETTER AFTER YOU HAVE COVERED : ARRAYS, LOOPS AND FUNCTIONS ******
// ****** THE ONLY OBJECTIVE OF WRITING THIS FUNCTION WAS TO OBTAIN THE BINARY REPRESENTATION OF DECIMAL INTEGERS SO THAT BIT-WISE AND-ing, OR-ing, COMPLEMENT AND BIT-SHIFTING COULD BE UNDERSTOOD WITH GREAT EASE ******

void PrintBinaryFormOfNumber(unsigned int decimal_number)
{
    //variable declarations
    unsigned int quotient, remainder;
    unsigned int num;
    unsigned int binary_array[8];
    int i;

    //code
    for (i = 0; i < 8; i++)
        binary_array[i] = 0;

    printf("The Binary Form Of The Decimal Integer %d Is\t=\t", decimal_number);
    num = decimal_number;
    i = 7;
    while (num != 0)
    {
        quotient = num / 2;
        remainder = num % 2;
        binary_array[i] = remainder;
        num = quotient;
        i--;
    }

    for (i = 0; i < 8; i++)
        printf("%u", binary_array[i]);

    printf("\n\n");
}

```