```
1 #include <stdio.h>
 2
 3 int main(void)
 4 {
 5
        //function prototypes
 6
       void PrintBinaryFormOfNumber(unsigned int);
 7
       //variable declarations
 8
 9
       unsigned int a;
10
       unsigned int b;
11
       unsigned int result;
12
13
       //code
14
       printf("\n\n");
15
       printf("Enter An Integer = ");
16
       scanf("%u", &a);
17
18
       printf("\n\n");
       printf("Enter Another Integer = ");
19
20
       scanf("%u", &b);
21
       printf("\n\n\n\n");
22
23
       result = a ^ b;
       printf("Bitwise XOR-ing Of \nA = %d (Decimal) and B = %d (Decimal) gives
24
          result %d (Decimal).\n\n", a, b, result);
25
26
       PrintBinaryFormOfNumber(a);
27
       PrintBinaryFormOfNumber(b);
28
       PrintBinaryFormOfNumber(result);
29
30
       return(0);
31 }
32
33
34 // ***** BEGINNERS TO C PROGRAMMING LANGUAGE : PLEASE IGNORE THE CODE OF THE
     FOLLOWING FUNCTION SNIPPET 'PrintBinaryFormOfNumber()' ******
35 // ****** YOU MAY COME BACK TO THIS CODE AND WILL UNDERSTAND IT MUCH BETTER AFTER ➤
     YOU HAVE COVERED : ARRAYS, LOOPS AND FUNCTIONS ******
36 // ***** 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 ******
37
38 void PrintBinaryFormOfNumber(unsigned int decimal_number)
39 {
40
        //variable declarations
41
       unsigned int quotient, remainder;
42
       unsigned int num;
43
       unsigned int binary_array[8];
44
       int i;
45
46
       //code
       for (i = 0; i < 8; i++)
47
```

```
\underline{\dots\text{-} \texttt{Operators} \backslash \texttt{04-BitwiseOperators} \backslash \texttt{03-BitwiseXOR} \backslash \texttt{BitwiseXOR.c}}
```

```
2
```

```
\overline{binary\_array[i] = 0};
48
49
        printf("The Binary Form Of The Decimal Integer %d Is\t=\t", decimal_number);
50
51
        num = decimal_number;
52
        i = 7;
53
        while (num != 0)
54
55
            quotient = num / 2;
56
            remainder = num % 2;
57
            binary_array[i] = remainder;
58
            num = quotient;
59
            i--;
60
        }
61
62
        for (i = 0; i < 8; i++)</pre>
63
            printf("%u", binary_array[i]);
64
        printf("\n\n");
65
66 }
67
```