

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      //function prototypes
6      void PrintBinaryFormOfNumber(unsigned int);
7
8      //variable declarations
9      unsigned int a;
10     unsigned int num_bits;
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");
19     printf("By How Many Bits Do You Want To Shift A = %d To The Right ? ", a);
20     scanf("%u", &num_bits);
21
22     printf("\n\n\n");
23     result = a >> num_bits;
24     printf("Bitwise RIGHT-SHIFTing A = %d By %d Bits \nGives The Result = %d\n\n", a, num_bits, result);
25     PrintBinaryFormOfNumber(a);
26     PrintBinaryFormOfNumber(result);
27
28     return(0);
29 }
30
31
32 // ***** BEGINNERS TO C PROGRAMMING LANGUAGE : PLEASE IGNORE THE CODE OF THE FOLLOWING FUNCTION SNIPPET 'PrintBinaryFormOfNumber()' *****
33 // ***** YOU MAY COME BACK TO THIS CODE AND WILL UNDERSTAND IT MUCH BETTER AFTER YOU HAVE COVERED : ARRAYS, LOOPS AND FUNCTIONS *****
34 // ***** 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 *****
35
36 void PrintBinaryFormOfNumber(unsigned int decimal_number)
37 {
38     //variable declarations
39     unsigned int quotient, remainder;
40     unsigned int num;
41     unsigned int binary_array[8];
42     int i;
43
44     //code
45     for (i = 0; i < 8; i++)
46         binary_array[i] = 0;
47
```

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