2 The Binary Representation

2.1 Reading a Character Using `getchar`

You can read a character using `getchar`. Though the name of the function indicates `getchar` returns a `char` type, it returns `int` value. Owing to this property, you may use arithmetic operators on the return value of `getchar`. The following shows a sample program to return the ASCII and numeric values of a decimal digit:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int c = getchar(), d = c - '0';
6
7      printf("ASCII('%c')   = %d\n", c, c);
8      printf("Decimal('%c') = %d\n", c, d);
9      printf("LSB(%d): %d\n", d, d & 1);
10
11     return 0;
12 }
```

Note that `'0'` in Line 5 denotes the character literal of the decimal digit 0. Since the `char` is a subtype of `int`, you may subtract `'0'` from `c` which contains an `int` value.

From Line 9, you can find another use of the operator `&`. Though the operator `&` in a unary form denotes the *address-of* operator, the same operator in a binary form denotes the bitwise-AND operator, returning 1 only for the case of 1s in both operands. Therefore, the expression `d & 1` extracts the LSB (the least significant bit) of `d`. You may use other bitwise operators such as `|` (the bitwise-OR) and `^` (the bitwise-XOR).

2.2 Programming Lab 2: bits.c

Read a decimal integer n of two digits and print the decimal value of the binary number consisting of the 7th, 5th, 3rd, and 1st bits of the binary representation of n. The bit places are counted from the LSB, i.e. the LSB is the first bit of the binary representation.

Given $n = 75$, for example, the binary representation of n is 1001011. Since the 7th, 5th, 3rd, and 1st bits are 1, 0, 0, and 1, respectively, your program should print 9. If you cannot find any of the bit places of the binary representation of n, you should assume that the corresponding bit is 0 since it really holds for numbers, say, 1100 is equal to 0001100.

The input consists of a single line containing a decimal integer n of two decimal digits ($9 < n < 100$). Your program should print the number consisting of odd bit places of the binary representation of n to the standard output.

Additional requirements for bonus points: Your program file should be named as `bits.c` as noted in the title of this section. Do not use `scanf`, but use `getchar`.

Input	Output
12	2
75	9
49	5