Programming Practice: Fibonacci Sequence

1. A recursive Fibonacci function is defined as below:

$$f(n)=\begin{cases}1 & \text{if } n=0\\ 1 & \text{if } n=1\\ f(n-2)+f(n-1) & \text{if } n>1\end{cases}$$

Another fast recursive Fibonacci sequence is defined as below:

$$f(n)=\begin{cases}1 & \text{if } n=0\\ 1 & \text{if } n=1\\ 2 & \text{if } n=2\\ f\left(\frac{n}{2}+1\right)\times f\left(\frac{n}{2}\right)-f(\frac{n}{2}-1)\times f(\frac{n}{2}-2) & \text{if } n>2 \wedge n \text{ is even}\\ f\left(\frac{n}{2}+1\right)\times f\left(\frac{n}{2}\right)+f\left(\frac{n}{2}\right)\times f\left(\frac{n}{2}-1\right) & \text{if } n>2 \wedge n \text{ is odd}\end{cases}$$

Write a C program to implement three functions: recursive Fibonacci function, iterative Fibonacci function, and fast recursive Fibonacci function. Then, repeatedly read an integer n. If n is greater than or equal to 0, compute and output Fibonacci number f(n) using three Fibonacci functions; otherwise, stop the program. Program solution: fibonacci.c.

2. A Fibonacci number is defined by the following function:

$$fib(n)\equiv\begin{cases}1 & \text{if } n=0\\ 1 & \text{if } n=1\\ fib(n-1)+fib(n-2) & \text{if } n>1\end{cases}$$

The following table is the Fibonacci number of n, for $0\le n\le 10$:

n	0	1	2	3	4	5	6	7	8	9	10
$fib(n)$	1	1	2	3	5	8	13	21	34	55	89

Any positive integer can be represented as the sum of Fibonacci numbers without duplicate elements, e.g., 19=8+5+3+2+1=13+5+1 and 30=13+8+5+3+1=21+8+1. The representation is *not unique*. However, if we require ***no*** two consecutive Fibonacci sequence elements are allowed in the sum, then the representation is ***unique***. That is, 19=13+5+1 and 30=21+8+1. Let use use 0 and 1 to mark the occurrence of Fibonacci numbers to define ***Fibonacci numerals***, such as

$19=13\times1+8\times0+5\times1+3\times0+2\times0+1\times1=101001_{\text{fib}}$ and

$30=21\times1+13\times0+8\times1+5\times0+3\times0+2\times0+1\times1=1010001_{\text{fib}}$.

Write a C program to input a positive integer n, for $0<n\le100000000$, and compute and output the Fibonacci numeral of integer n. Repeat the program until n equals to 0. Program solution: fibonacci_numeral.c. Program execution example:

```
D:\>fibonacci_numeral
Enter a positive integer n between 1 and 100000000 (0 to stop): 1
The Fibonacci numeral of n is: 1 (fib)
---------------------------------------
Enter a positive integer n between 1 and 100000000 (0 to stop): 5
The Fibonacci numeral of n is: 1000 (fib)
---------------------------------------
Enter a positive integer n between 1 and 100000000 (0 to stop): 19
The Fibonacci numeral of n is: 101001 (fib)
---------------------------------------
Enter a positive integer n between 1 and 100000000 (0 to stop): 30
The Fibonacci numeral of n is: 1010001 (fib)
---------------------------------------
Enter a positive integer n between 1 and 100000000 (0 to stop): 100
The Fibonacci numeral of n is: 1000010100 (fib)
---------------------------------------
Enter a positive integer n between 1 and 100000000 (0 to stop): 500
The Fibonacci numeral of n is: 1001010000000 (fib)
---------------------------------------
Enter a positive integer n between 1 and 100000000 (0 to stop): 0

D:\>
```