

Recursive function:

程式碼:

```
int Fibonacci_recursion(int i){
    if(i==0) return 0;
    if(i==1) return 1;
    return Fibonacci_recursion(i-1)+Fibonacci_recursion(i-2);
};
```

說明:

根據輸入的值，這個函式會由 top-down 的方式(由 i 到 1)遞迴，直到終止條件 i==1 或 i==0 (同時，如果輸入的值為 1 or 0 時，則直接輸出結果)

Iterative function:

程式碼:

```
int Fibonacci_iterative(int i){
    if (i==0) return 0;
    if (i==1) return 1;
    int f0=0,f1=1,fn=0;
    for (int j=1;j<i;j++){
        fn=f1+f0;
        f0=f1;
        f1=fn; //for next loop calculation
    }
    return f1;
};
```

說明:

根據輸入的值，此迭代函式會由 bottom up 的方式，自 f0=0, f1=1 逐漸累加到指定的層數。為了避免呼叫太多變數，每一次的 loop 都會更新 f0, f1 的值，供下個 loop 使用，並在最後輸出 f1(因為 f1 的值最後更新為 fn)。