Maintenance: Let the loop invariant hold just before the iteration, when $i = l$, $l \leq n$. $\therefore f\_i = F(l-2)$ and $f\_j = F(l-1)$.
On line 12, we have $f\_K = F(l-2) + F(l-1) = F(l)$, the lth Fibonacci no.
On line 13, we initialize f_i to $F(l-1)$ and f_j to $F(l)$. in line 14.
Now, the loop index i gets updated to $l+1$. The loop invariant holds
$\because f\_j = F((l+1)-1) = F(i-1)$ and $f\_i = F(l-1) = F((l+1)-2) = F(i-2)$

Termination: The loop terminates when $i = n+1$. Using the maintenance condn, $f\_i = F((n+1)-2) = F(n-1)$, $f\_j = F((n+1)-1) = F(n) = f\_K$, which is the nth Fibonacci no. $\square$

9) Write an iterative function to compute a binomial coefficient, then transform it into an equivalent recursive function.

Binomial coefficient: ${}^nC_K = \frac{n!}{(n-K)!\,K!}$, $n \geq K$

Recursive formulation: $\boxed{{}^nC_K = {}^{n-1}C_K + {}^{n-1}C_{K-1}}$

```
long iterative_binomial_coefficient (int n, int K)
{
  return iterative_factorial(n)/(iterative_factorial(n-K) * iterative_factorial(K));
}
```

Proof Sketch: Since we have already verified the correctness of the iterative factorial function, and since the binomial coefficient is defined as $\boxed{{}^nC_K = \frac{n!}{(n-K)!\,K!}}$ the iterative function correctly computes the value of ${}^nC_K$ for all $0 \leq K \leq n$. Hence the function is correct (Provided factorial results are computed correctly)

```
1. long recursive_binomial_coefficient (int n, int K)
2. {
3.   if (K==0)
4.     return 1;
5.   else if (n==K)
6.     return 1;
7.   else return recursive_binomial_coefficient (n-1, K)
       + recursive_binomial_coefficient (n-1, K-1);
8. }
```