public static voidFibonacci\_Iterative(int len)

{

int a = 0, b = 1, c = 0;

Console.Write("{0} {1}", a,b);

for (int i = 2; i < len; i++)

{

c= a + b;

Console.Write(" {0}", c);

a= b;

b= c;

}

}

Input: Fibonacci\_Iterative(9);

Output: 0 1 1 2 3 5 8 13 21

Press any key to continue . . .

public static voidFibonacci\_Recursive(int len)

{

Fibonacci\_Rec\_Temp(0, 1, 1, len);

}

private static voidFibonacci\_Rec\_Temp(int a, int b, int counter, int len)

{

if (counter <= len)

{

Console.Write("{0} ", a);

Fibonacci\_Rec\_Temp(b, a + b, counter+1, len);

}

}

Input: Fibonacci\_Recursive(11);

Output: 0 1 1 2 3 5 8 13 21 34 55

Press any key to continue . . .

public static int GetNthFibonacci\_Ite(int n)

{

int number = n - 1; //Need to decrement by 1 since we are starting from 0

int[] Fib = new int[number + 1];

Fib[0]= 0;

Fib[1]= 1;

for (int i = 2; i <= number;i++)

{

Fib[i] = Fib[i - 2] + Fib[i - 1];

}

return Fib[number];

}

Input: GetNthFibonacci\_Ite(4);

Output: 2

Press any key to continue . . .

public static int GetNthFibonacci\_Rec(int n)

{

if ((n == 0) || (n == 1))

{

return n;

}

else

returnGetNthFibonacci\_Rec(n - 1) + GetNthFibonacci\_Rec(n - 2);

}

Input: GetNthFibonacci\_Rec(8-1);

Output: 13

Press any key to continue . . .