Assignment: 03 Name: Methew Guda
Due Date: 03-11-2015 A#: A00381751

Q 01.

$F(0) = 0$
$F(1) = 1$
$F(2) = F(2 - 1) + F(2 - 2) = F(1) + F(0) = 1 + 0 = 1$: 2^0 Addition & 2^1 Subtractions
$F(3) = F(3 - 1) + F(3 - 2)$
$= F(2) + F(1)$
$= F(2 - 1) + F(2 - 2) + F(1) = F(1) + F(0) + F(1) = 1 + 0 + 1 = 2$: 2^1 Additions & 2^2 Subtractions
$F(4) = F(4 - 1) + F(4 - 2)$
$= F(3) + F(2)$
$= F(3 - 1) + F(3 - 2) + F(2 - 1) + F(2 - 2)$
$= F(2) + F(1) + F(1) + F(0)$
$= F(2 - 1) + F(2 - 2) + F(1) + F(1) + F(0)$
$= F(1) + F(0) + F(1) + F(1) + F(0) = 1 + 0 + 1 = 2$: 2^2 Additions & 2^3 Subtractions
.
.
.
.
$F(N) = F(N - 1) + F(N - 2) = 2^{n-2}$ additions & 2^{n-1} Subtractions

Characteristic Polynomial:

$C(N) = C(N - 1) + C(N - 2)$ ---------------------------(1)
$C(N - 1) = C(N - 2) + C(N - 3)$ ----------------------(2)

By (1) – (2)

$C(N) - C(N - 1) = C(N - 1) + C(N - 2) - C(N - 2) - C(N - 3)$
$C(N) - C(N - 1) = C(N - 1) - C(N - 3)$
$C(N) - 2C(N - 1) + C(N - 3) = 0$
$X^n - 2X^{(n-1)} + X^{(n-3)} = 0$
$X^3 - 2X^2 + 1 = 0$

$$\begin{vmatrix} 1 & (1 - 5)/2 & (1 + 5)/2 \\ 1 & (1 - 5)^2/4 & (1 + 5)^2/4 \\ 1 & (1 - 5)^3/8 & (1 + 5)^3/8 \end{vmatrix} \begin{vmatrix} C_1 \\ C_2 \\ C_3 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix}$$

If N = 10 then the number of calculations will be,

Additions: $2^{10-2} = 2^8 = 256$
Subtractions: $2^{10-1} = 2^9 = 512$

Q 02. Iterative Fibonacci Function:

```
int Fibonacci (N)
  firstNumber = 0, secondNumber = 1, result = 0

  if (N == 0) return 0;
  if (N == 1) return 1;

  for (i = 2;  i <= N;  i++)
     result = firstnumber + secondnumber;
     firstnumber = secondnumber;
     secondnumber = result;
   end for

  return result;
end Fibonacci
```

Iterative version of Fibonacci function does not have any calculations on subtractions and multiplications. Also there is no addition calculation for N = 0 and N = 1. So the only calculations will additions and for N, the number of additions it will be n – 2.

Now for N = 10: There is no calculations in iterative Fibonacci when N = 0 and N = 1. The calculations start when N = 2. Fibonacci function then add 1st number (0) and 2nd number (1), put it into a variable. Replace the value of 1st number with 2nd number and 2nd number with the summed number (0 + 1 = 1). Iterative Fibonacci function repeat this process until N = 10.

	N(0)	N(1)	N(2)	N(3)	N(4)	N(5)	N(6)	N(7)	N(8)	N(9)
	0	1	1	2	3	5	8	13	21	34
Additions	0	0	1	1	1	1	1	1	1	1
Subtractions	0	0	0	0	0	0	0	0	0	0
Multiplications	0	0	0	0	0	0	0	0	0	0

So from the table we can see:
Additions: 8
Subtractions: 0
Multiplications: 0

Q 03. Nth element of the Fibonacci sequence

```
int NthElement (N)

  if (N == 0 || N == 1) return N;
   else
     result = ((1 + 5^(1/5))^n – (1 – 5^(1/5))^n) / (2^n * 5^(1/5))
   end if

  return result;
end NthElement
```

After analyzing the closed form expression, which is $((1 + 5^{(1/5)})^n – (1 – 5^{(1/5)})^n) / (2^n * 5^{(1/5)})$, there will be **(n-1)** times additions because of $(1 + 5^{(1/5)})^n$. The will be **(n-1) + 1 = n** times subtractions, because (n-1) times for $(1 – 5^{(1/5)})^n$ and extra 1 time for subtracting $(1 – 5^{(1/5)})^n$ from $(1 + 5^{(1/5)})^n$. There will be **3(n-1) + 1 = 3n - 2** times multiplications, because (n-1) times for $(1 + 5^{(1/5)})^n$, (n-1) times for $(1 + 5^{(1/5)})^n$, (n-1) times for 2^n and extra 1 time for multiplying 2^n with $5^{(1/5)}$.

So now if N = 10, then the number of Calculations will be:

Addition: (10 - 1) = 9 times
Subtractions: (10 – 1) + 1 = 10 times
Multiplications: 3(10 -1) + 1 = 28 times

Q 04.

```
using System;

namespace FibonacciRecursive
{
    class Program
    {
        private static int addCounterRecursive;
        private static int subCounterRecursive;
        private static int mulCounterRecursive;

        static int Fibonacci(int n)
        {
            if (n < 2) return n;
            else
            {
                addCounterRecursive += 1;
                subCounterRecursive += 2;
                return (Fibonacci(n - 2) + Fibonacci(n - 1));
            }
        }


        static void Main(string[] args)
        {
            Console.Write("Enter the length of the Fibonacci Series: ");
            int length = Convert.ToInt32(Console.ReadLine());


            Console.Write("\nRecursive Fibonacci Series: ");

            for (int i = 0; i < length; i++)
            {
                Console.Write("{0} ", Fibonacci(i));
            }

            Console.WriteLine("\nRecursive Additions: {0}", addCounterRecursive);
            Console.WriteLine("Recursive Subtractions: {0}", subCounterRecursive);
            Console.WriteLine("Recursive Multiplications: {0}", mulCounterRecursive);

            Console.ReadKey();
        }
    }
}
```

Output:

```
Enter the length of the Fibonacci Series: 10

Recursive Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
Recursive Additions: 133
Recursive Subtractions: 266
Recursive Multiplications: 0
```

Q 05.

```
using System;

namespace FibonacciIterative
{
    class Program
    {
        private static int addCounterIterative;
        private static int subCounterIterative;
        private static int mulCounterIterative;

        static void Fibonacci(int n)
        {
            int firstnumber = 0, secondnumber = 1, result = 0;

            if (n == 0)
            {
                Console.WriteLine("{0}", firstnumber);
            }
            if (n == 1)
            {
                Console.WriteLine("{0} {1}", firstnumber, secondnumber);
            }
            else
            {
                Console.Write("{0} ", firstnumber);
                for (int i = 2; i < n; i++)
                {
                    addCounterIterative++;
                    result = firstnumber + secondnumber;
                    Console.Write("{0} ", result);
                    firstnumber = secondnumber;
                    secondnumber = result;
                }
            }
        }
```

```
        static void Main(string[] args)
        {
            Console.Write("Enter the length of the Fibonacci Series: ");
            int length = Convert.ToInt32(Console.ReadLine());

            Console.Write("\nIterative Fibonacci Series: ");

            Fibonacci(length);

            Console.WriteLine("\nIterative Additions: {0}", addCounterIterative);
            Console.WriteLine("Iterative Subtractions: {0}", subCounterIterative);
            Console.WriteLine("Iterative Multiplications: {0}", mulCounterIterative);

            Console.ReadKey();
        }
    }
}
```

Output:

```
Enter the length of the Fibonacci Series: 10

Iterative Fibonacci Series: 0 1 2 3 5 8 13 21 34
Iterative Additions: 8
Iterative Subtractions: 0
Iterative Multiplications: 0
```

Q 06.

```
using System;

namespace NthFibonacciElement
{
    class Program
    {
        private static int addCounterNthElement;
        private static int subCounterNthElement;
        private static int mulCounterNthElement;

        static int NthElement(int n)
        {
            double top, bottom, result;

            if ((n == 0) || (n == 1))
            {
                return n;
            }
            else
            {
                addCounterNthElement = (n - 1);
                subCounterNthElement = (n - 1) + 1;
                mulCounterNthElement = (3 * n) - 2;
                n = n - 1;
                top = (Math.Pow((1 + Math.Sqrt(5)), n)) - (Math.Pow((1 - Math.Sqrt(5)), n));
                bottom = (Math.Pow(2, n)) * (Math.Sqrt(5));

                result = top / bottom;

                return (int)result;
            }
        }
```

```csharp
        static void Main(string[] args)
        {

            Console.Write("Enter the nth number of the Fibonacci Series: ");
            int value = Convert.ToInt32(Console.ReadLine());

            Console.Write("\n{0}th Fibonacci Element: {1}", value, NthElement(value));

            Console.WriteLine("\nNth Element Additions: {0}", addCounterNthElement);
            Console.WriteLine("Nth Element Subtractions: {0}", subCounterNthElement);
            Console.WriteLine("Nth Element Multiplications: {0}", mulCounterNthElement);

            Console.ReadKey();
        }

    }
}
```

Output:

```
Enter the nth number of the Fibonacci Series: 10

10th Fibonacci Element: 34
Nth Element Additions: 9
Nth Element Subtractions: 10
Nth Element Multiplications: 28
```

Q 07.

My experimental counts compare with algorithm analysis is same except for the recursive case. In recursion, it shows that I was suppose to have 256 additions and 512 subtractions but the programs results show different. My implementation was robust. I calculated all the 10 elements of Fibonacci series including 0^{th} element. N = 48 is the highest value that the methods can compute. When I put 49 I got garbage value on output and time increased when recursion is performed. In conclusion I want to state that recursion is not useful for Fibonacci when N is higher.