**Name:**

**Java Programming**

**Lab Exercise 12/8/2021**

**Recursion**

In each of the following recursion problems, state what’s printed.

1. System.out.println( rig(4) );

public static int rig(int n)

{

if ( (n = = 0) )

{

return 5;

}

else if ( n = = 1)

{

return 8;

}

else

{

return rig(n – 1) - rig(n – 2);

}

}

1. System.out.println( mm(6) );

public static int mm(int n)

{

if (n<=0)

return 10;

else

return n + mm(n-1);

}

1. System.out.println( adrml(5) );

public static int adrml(int n)

{

if (n<=1)

return n;

else

return n \* adrml(n-2);

}

1. System.out.println(bud(1));

public static int bud(int n)

{

if (n>5)

return n - 2;

else

return n + bud(n +1);

}

1. System.out.println(zing(0));

public static int zing(int n)

{

if (n > 10)

return n - 2;

else

{

n = n \* 3;

return n + zing(n + 2);

}

}

1. crch(12);

public static void crch(int n)

{

if (n <= 0)

System.out.print(n);

else

{

crch(n / 3);

System.out.print(“,” + n);

}

}

1. elvis(11);

public static void elvis(int n)

{

if (n <= 3)

System.out.print(n + 1);

else

{

elvis(n-3);

System.out.print(“>>” + (n – 1));

}

}

1. sal(5);

public static int sal(int n)

{

if (n = = 2)

{

return 100;

}

else if (n = = 3)

{

return 200;

}

else

{

return (2 \* sal(n - 1) + sal(n - 2) + 1);

}

}

1. puf(4);

public static void puf(int n)

{

if(n = = 1)

{

System.out.print(“x”);

}

else if( n%2 = = 0) //n is even

{

System.out.print(“{”);

puf(n-1);

System.out.print(“}”);

}

else //n is odd

{

System.out.print(“<”);

puf(n-1);

System.out.print(“>”);

}

}

1. bc(6, 2);

public static void bc(int p, int q)

{

if (p/q = = 0)

{

System.out.println(p + q + 1);

}

else

{

System.out.println(p);

bc(p/q, q);

}

}

**Project … Factorial**

You are to write a recursive factorial function. Your function should return n! of any value n. Remember 0! = 1.

**Project… Fibonacci**

You are to write a recursive fibonacci function that returns the kth Fibonacci number.

**Project… ModFibonacci**

You are to write a recursion routine to generate the kth term of a “modified” Fibonacci sequence.

Our modified sequence will be defined as follows:

modFibonacci(0) = 3

modFibonacci(1) = 5

modFibonacci(2) = 8

modFibonacci(n) = modFibonacci(n - 1) + modFibonacci(n -2) + modFibonacci(n-3)

For your convenience several terms of this sequence are:

3 5 8 16 29 53 98 . . .

Call your new class *ModFib* and create a *static* method in it called *modFibonacci*.

Test your new class with the following *Tester* class:

import java.io.\*;

import java.util.\*;

public class Tester

{

public static void main(String args[])

{

Scanner kbReader = new Scanner(System.in);

System.out.print("Generate which term number? ");

int k = kbReader.nextInt( );

System.out.println("Term #" + k + " is " + ModFib.modFibonacci(k));

}

}

Typical runs should look like this:

Generate which term number? 5

Term #5 is 53

Generate which term number? 6

Term #6 is 98