Computer Science 221 Lab 9 Worth: 6 pts. Assigned: Monday, April 07, 2014

[Adopted from a lab developed by Dr. Ed McDowell]

The binomial coefficient C(n,k) is defined recursively for nonnegative integers n and k, where 0 <= k <= n, as follows:

C(n,k) = 1 if k = 0 or k = n.

Otherwise, C(n,k) = C(n-1,k-1) + C(n-1,k).

The following table, known as Pascal's Triangle, lists the first few binomial coefficients.

Binomial Coefficient C(n,k)

k=0 k=1 k=2 k=3 k=4 k=5 k=6 k=7 k=8 k=9 k=10

n=0 1

n=1 1 1

n=2 1 2 1

n=3 1 3 3 1

n=4 1 4 6 4 1

n=5 1 5 10 10 5 1

n=6 1 6 15 20 15 6 1

n=7 1 7 21 35 35 21 7 1

n=8 1 8 28 56 70 56 28 8 1

n=9 1 9 36 84 126 126 84 36 9 1

n=10 1 10 45 120 210 252 210 120 45 10 1

1. The first version of the program prints Pascal's Triangle (without the title)

a) Compile and run the following program.

// Lab 9: Your name

public class Lab9a {

public static void main(String[] args) {

int n, k, coef;

for (n = 0; n <= 10; ++n) {

for (k = 0; k <= n; ++k) {

coef = binCoef(n,k);

System.out.printf("%5d", coef);

}

System.out.printf("%n");

}

}

public static int binCoef(int n, int k) {

// Use the recursive definition above to

// compute and return C(n,k).

}

}

b) Now modify the program so that it prints the title and the left side.

Print output here:

|  |
| --- |
|  |

2. Modify the program to print the number of recursive calls needed to compute C(n,k). Count the calls in a class variable as shown. Clear the count before making the initial call. Increment the count each time the recursive function is called. Compile and run the program. Hand in a listing of the Java source code and the output from a successful run.

a) Modify the code below so that there is an appropriate title [i.e. # of calls]

// Lab 9: Your name

public class Lab9b {

public static int calls;

public static void main(String[] args) {

int n, k;

for (n = 0; n <= 10; ++n) {

for (k = 0; k <= n; ++k) {

calls = 0;

binCoef(n,k);

System.out.printf("%5d", calls);

}

System.out.printf("%n");

}

}

public static int binCoef(int n, int k) {

++calls;

// Your code from item 1 above.

}

}

Put output here:

|  |
| --- |
|  |

b) Interpret the results:

How many calls are needed to find

(i) C(8,2) \_\_55\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ii) C(7,3) \_\_\_\_\_69\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(iii) C(10,4) \_\_\_419\_\_\_\_\_\_\_\_\_\_\_\_\_

**Please hand in this document along with a zipped file of your program.**