COSC 3327.01 Homework #1
Due Friday September 7th 11:55pm
You may submit this document online via canvas or in person (in class or dropping off at my office)

Determine the complexity of the following pseudocode snippets in both big O notation, and the overall operation count:

1)

```
result = 0
for (num in some_nums)
        result += num
return result
```

O(n)
Operation count: 1+2n+1

2)

```
int temp = 0
for (i =0; i < 100000000; i++)
        for (j = 0; j < N; j++)
                temp = 0
                while(temp < j)
                {
                        temp++;
                }
```

O(n^2)
Operation count: 1+1+100000000+100000000+1+n+n+1+n or 200000004 + 3n

3)

```
input: array A
L = 0
R = n − 1
while L <= R:
        m = floor((L + R) / 2)
        if A[m] < T:
                L = m + 1
        else if A[m] > T:
                R = m - 1
        else:
                return m
return unsuccessful
```

O(log(n))
1+1+1+log(n)+1 or 4+log(n)

Construct an appropriate algorithm to solve the following two problems.

Given two arrays containing sorted integers, create an efficient algorithm that generates a new array containing only elements which are in **both** arrays. For this algorithm you may assume that no element appears more than once inside a single array (no duplicates within an array).

```
index1=0;
index2=0;
size1 = A1.size();
size2=A2.size();
while(index1 < size1)
{ while(index2 < size2)
  { if(A1[index1] == A2[index2])
    { A3.add(A1[index1]);
    } else if(A1[index1] > A2[index2])
    { index2++;
    } else // if A1 at index1 is less than A2 at index2
    { break;
    }
  }
 index1++;
}
```

Create an algorithm which takes as input two distinct integers int1 and int2 and returns the product of the two using only addition. Your solution may be iterative or recursive.

```
static int product(int x, int y)
{
  int retVal;
  if(x == 1)
  { retVal = y;
  } else if(y==1)
  { retVal =  x;
  } else if(x > y || x == y)
  { retVal = x + product(x, y-1);
  } else if(x < y)
  { retVal = y + product(x-1, y);
  }
  return retVal;
```

Solve

What is the minimum number of multiplications needed to generate x^{60} when starting from x? (you may only use terms of x to solve this problem)

x_3 = x*x*x;

x_6 = x_3*x_3;

x_12 = x_6*x_6;

x_24=x_12*x_12;

x_48=x_24*x_24;

x_60 = x_48*x_12;

The answer is 7.

Bonus: Construct a program that will generate a set of 50 random integers. Your program should output the generated integers, and then output the min, max, and average values of the set. What is the overall runtime of your program in terms of total operations? What is the complexity of your program

```
public static void randomIntArray()
{
    Random r = new Random();
    int n = 50;
    int[] array = new int[n];

    for(int i = 0; i < n; i++)
    {
        array[i] = r.nextInt();
    }
    int min = array[0];
    int max = array[0];
    int sum = 0;
    for(int j = 0; j < n; j++)
    {
        sum += array[j];
        if(array[j] > max)
        {
            max = array[j];
        }
        else if(array[j] < min)
        {
            min = array[j];
        }
        System.out.println(" " + array[j]);
    }
    int average = sum / n;
    System.out.println("Min: " + min + " Max: " + max + " Average: " + average);
}
```

The number of operations is 3+1+2n+n+3+1+2n+n+n+1+1 or 10+7n and the complexity is just O(n)