1 Short Answer (10 Points Each)

1. Write a method that takes in an array of doubles as a parameter and outputs the average of the elements in the array. If the array is of zero length then the method should return 0.

   Solution:

```
public static double average(double A[]) {
    double avg = 0;
    if (A.length > 0) {
        double sum = 0;
        for (int i = 0; i < A.length; i++)
            sum += A[i];
        avg = sum / A.length;
    }
    return avg;
}
```

2. Write a method that will take in an array of doubles as a parameter and outputs the largest value in the array. If the array is of zero length then the method should return 0. You may not assume anything about the values being stored in the array.

   Solution:

```
public static double maximum(double A[]) {
    if (A.length == 0)
        return 0;

    double max = A[0];
    for (int i = 1; i < A.length; i++)
        if (max < A[i])
            max = A[i];

    return max;
}
```

3. Write a method that will take in an array of longs as a parameter. The method will leave the first two entries of the array alone and alter the remaining entries of the array by replacing the array entry with the sum of the two entries before it. For example, if the input array is

2	3	3	2	7	6	4	7	5

   the result will be

2	3	5	8	13	21	34	55	89

   Solution:

```
public static void fib(long A[]) {
    for (int i = 2; i < A.length; i++)
        A[i] = A[i - 1] + A[i - 2];
}
```

4. Write a linear search method for an integer array that takes in an array and target value as parameters and returns the first position of the target in the array. If the target is not in the array then the method should return -1.

   Solution:

```
public static int linearSearch(int[] A, int N) {
    for (int index = 0; index < A.length; index++) {
        if (A[index] == N)
            return index;
    }
    return -1;
}
```

5. Write a method that does either the bubble sort, insertion sort or selection sort for an array of integers. You must state which sort you are writing.

   Solution:

```
public static void BubbleSort(int A[]) {
    for (int i = A.length - 1; i > 0; i--) {
        for (int j = 0; j < i; j++) {
            if (A[j] > A[j + 1]) {
                int temp = A[j];
                A[j] = A[j + 1];
                A[j + 1] = temp;
            }
        }
    }
}

public static void insertionSort(int[] A) {
    for (int itemsSorted = 1; itemsSorted < A.length; itemsSorted++) {
        int temp = A[itemsSorted];
        int loc = itemsSorted - 1;
        while (loc >= 0 && A[loc] > temp) {
            A[loc + 1] = A[loc];
            loc = loc - 1;
        }
        A[loc + 1] = temp;
    }
}

public static void selectionSort(int[] A) {
    for (int lastPlace = A.length - 1; lastPlace > 0; lastPlace--) {
        int maxLoc = 0;
        for (int j = 1; j <= lastPlace; j++)
            if (A[j] > A[maxLoc])
                maxLoc = j;

        int temp = A[maxLoc];
        A[maxLoc] = A[lastPlace];
        A[lastPlace] = temp;
    }
}
```

2 Program Traces (20 Points Each)

1. Write the output of the program.

```
1  public class Exam03Trace1 {
2
3      public static void PrintArray(int A[]) {
4          for (int i = 0; i < A.length; i++) {
5              System.out.printf("%4d", A[i]);
6          }
7          System.out.println();
8      }
9
10     public static void DoSomething1(int A[]) {
11         int lastent = 0;
12         for (int i = 0; i < A.length; i++) {
13             A[i] = A[i] + lastent;
14             lastent = A[i];
15         }
16     }
17
18     public static void DoSomething2(int A[]) {
19         for (int i = 0; i < A.length; i++) {
20             A[i] = A[A.length - 1 - i];
21         }
22     }
23
24     public static void main(String[] args) {
25         int A[] = new int[8];
26         int B[] = new int[8];
27
28         for (int i = 0; i < 8; i++)
29             A[i] = i + 1;
30
31         PrintArray(A);
32         DoSomething1(A);
33         PrintArray(A);
34         System.out.println("----------");
35
36         A[0] = 2;
37         for (int i = 1; i < 8; i++)
38             A[i] = 2 * A[i - 1] % 11;
39
40         PrintArray(A);
41         DoSomething2(A);
42         PrintArray(A);
43         System.out.println("----------");
44
45         A[0] = 10;
46         for (int i = 1; i < 8; i++)
47             A[i] = A[i - 1] - 3;
48
49         B = A;
50         PrintArray(A);
51         PrintArray(B);
52
53         B[3] = 20;
54         A[6] = 4;
55         A[0] = B[7];
56
57         PrintArray(A);
58         PrintArray(B);
59     }
60 }
```

Program Output

```
   1   2   3   4   5   6   7   8
   1   3   6  10  15  21  28  36
----------
   2   4   8   5  10   9   7   3
   3   7   9  10  10   9   7   3
----------
  10   7   4   1  -2  -5  -8 -11
  10   7   4   1  -2  -5  -8 -11
 -11   7   4  20  -2  -5   4 -11
 -11   7   4  20  -2  -5   4 -11
```

2. For each of the given inputs, write the output of the program.

```
1  import java.util.Scanner;
2
3  public class Exam03Trace2 {
4
5      public static void PrintArray(int A[]) {
6          for (int i = 0; i < A.length; i++) {
7              System.out.printf("%4d", A[i]);
8          }
9          System.out.println();
10     }
11
12     public static void MixArray1(int A[]) {
13         for (int i = 0; i < A.length; i++) {
14             A[i] = A[(i * i) % A.length];
15         }
16     }
17
18     public static void MixArray2(int A[]) {
19         int x = 3;
20         for (int i = 0; i < A.length; i++) {
21             int y = x * x % A.length;
22             int temp = A[x];
23             A[x] = A[y];
24             A[y] = temp;
25             x = y;
26         }
27     }
28
29     public static void main(String[] args) {
30         Scanner keyboard = new Scanner(System.in);
31         System.out.print("Array Size: ");
32         int ArrSz = keyboard.nextInt();
33
34         int A[] = new int[ArrSz];
35
36         System.out.print("Input elements: ");
37         for (int i = 0; i < ArrSz; i++)
38             A[i] = keyboard.nextInt();
39
40         System.out.print("Mix Number: ");
41         int mix = keyboard.nextInt();
42
43         PrintArray(A);
44
45         if (mix == 1)
46             MixArray1(A);
47         else
48             MixArray2(A);
49
50         PrintArray(A);
51         MixArray2(A);
52         PrintArray(A);
53     }
54 }
```

Program Output

```
Array Size: 5
Input elements: 1 2 3 4 5
Mix Number: 1
   1   2   3   4   5
   1   2   5   5   2
   1   5   5   2   2
```

Program Output

```
Array Size: 5
Input elements: 1 2 3 4 5
Mix Number: 2
   1   2   3   4   5
   1   4   3   5   2
   1   5   3   2   4
```

Program Output

```
Array Size: 6
Input elements: 2 -3 5 7 1 10
Mix Number: 1
   2  -3   5   7   1  10
   2  -3   1   7   1  -3
   2  -3   1   7   1  -3
```

3 Coding (20 Points)

The following is the start of a program that simulates the tossing of a coin and then reports the number of heads in the sequence of tosses, the longest string of consecutive heads, and number of times a head toss directly follows a tail toss, and the number HTTH toss sequences. For this simulation a 0 represents a tail toss and a 1 represents a heads toss. An example run is below. For each of the four methods that need to be written, write the appropriate code in the boxes on the next two pages.

```
1  import java.util.Scanner;
2
3  public class Exam03Code1 {
4
5      public static int CountHeads(int[] A) {
6          <<< Insert Code Here >>>
7      }
8
9      public static int CountHeadSequence(int[] A) {
10         <<< Insert Code Here >>>
11     }
12
13     public static int CountHeadAfterTail(int[] A) {
14         <<< Insert Code Here >>>
15     }
16
17     public static int CountHTTH(int[] A) {
18         <<< Insert Code Here >>>
19     }
20
21     public static void main(String[] args) {
22         Scanner keyboard = new Scanner(System.in);
23         System.out.print("Input the array size: ");
24         int arraySize = keyboard.nextInt();
25
26         int A[] = new int[arraySize];
27         for (int i = 0; i < A.length; i++)
28             A[i] = (int) (Math.random() * 2);
29
30         System.out.println("The number of heads = " + CountHeads(A));
31         System.out.println("The longest head sequence = " + CountHeadSequence(A));
32         System.out.println("The number of times a head follows a tail = " + CountHeadAfterTail(A));
33         System.out.println("The number of HTTH sequences = " + CountHTTH(A));
34     }
35 }
```

Example Program Run

```
Input the array size: 10000
The number of heads = 5094
The longest head sequence = 12
The number of times a head follows a tail = 2508
The number of HTTH sequences = 664
```

Solution:

```
1  import java.util.Scanner;
2
3  public class Exam03Code1 {
4
5      public static int CountHeads(int[] A) {
6          int count = 0;
7          for (int i = 0; i < A.length; i++)
8              if (A[i] == 1)
9                  count++;
10
11         return count;
12     }
13
14     public static int CountHeadSequence(int[] A) {
15         int count = 0;
16         int maxcount = 0;
17         for (int i = 0; i < A.length; i++)
18             if (A[i] == 1) {
19                 count++;
20                 if (count > maxcount)
21                     maxcount = count;
22             } else
23                 count = 0;
24
25         return maxcount;
26     }
27
28     public static int CountHeadAfterTail(int[] A) {
29         int count = 0;
30         for (int i = 1; i < A.length; i++)
31             if (A[i] == 1 && A[i - 1] == 0)
32                 count++;
33
34         return count;
35     }
36
37     public static int CountHTTH(int[] A) {
38         int count = 0;
39         for (int i = 3; i < A.length; i++)
40             if (A[i] == 1 && A[i - 1] == 0 && A[i - 2] == 0 && A[i - 3] == 1)
41                 count++;
42
43         return count;
44     }
45
46     public static void main(String[] args) {
47         Scanner keyboard = new Scanner(System.in);
48         System.out.print("Input the array size: ");
49         int arraySize = keyboard.nextInt();
50
51         int A[] = new int[arraySize];
52         for (int i = 0; i < A.length; i++)
53             A[i] = (int) (Math.random() * 2);
54
55         System.out.println("The number of heads = " + CountHeads(A));
56         System.out.println("The longest head sequence = " + CountHeadSequence(A));
57         System.out.println("The number of times a head follows a tail = " + CountHeadAfterTail(A));
58         System.out.println("The number of HTTH sequences = " + CountHTTH(A));
59     }
60 }
```