HW2

Question 1

```
/**
*  Gets a command-line argument (int), and prints all the divisors of the given number.
*/
public class Divisors {
  public static void main (String[] args) {
     int num = Integer.parseInt(args[0]);
     int i = 1;

     while(i<=num) {
        if (num%i == 0) {
           System.out.println(i);
        }i += 1;
     }
}
}
```

Question 2

```
/**
* Prints a given string, backward. Then prints the middle character in the string.
* The program expects to get one command-line argument: A string.
*/
public class Reverse {
  public static void main (String[] args){
     String sIn = args[0];
     String sOut = "";

     for (int i = sIn.length() - 1; i>=0; i--) {
        char c = sIn.charAt(i);
        sOut += c;
     }
     System.out.println(sOut);

     int midIndex;
     char midCharacter;
//checks if length of string is odd or even to determine where middle char is
     if (sIn.length()%2==0){
        midIndex = (sIn.length()/2) - 1;
     } else {
        midIndex = sIn.length()/2;
     }

     midCharacter = sIn.charAt(midIndex);
     System.out.println("The middle character is " + midCharacter);

  }
}
```

<u>Question 3</u>

```
/**
*  Generates and prints random integers in the range [0,10),
*  as long as they form a non-decreasing sequence.
*/
public class InOrder {
  public static void main (String[] args) {

    int Num1, Num2;

    Num1 = (int)(Math.random()*10);
      System.out.println(Num1);

    do {
      Num2 = (int)(Math.random()*10);
      if (Num1<=Num2){
        System.out.println(Num2);
        Num1 = Num2;
      }

    } while(Num1 <= Num2);


  }
}
```

Question 4

```
/**
*  Gets a command-line argument (int), and chekcs if the given number is perfect.
*/public class Perfect {
  public static void main (String[] args) {
    int Num = Integer.parseInt(args[0]);
    int i = 1;
    String sOut = Num + " is a perfect number since " + Num + " = ";

    int sum = 0;
    boolean hasDivisors = false;

    while(i < Num) {
      if (Num%i == 0) {
        if (hasDivisors){
          sOut += " + ";
        }
        sOut += i ;
        sum += i;
        hasDivisors = true;
      }
      i += 1;
    }

    if (sum==Num){
      System.out.println(sOut);
    } else {
      System.out.println(Num + " is not a perfect number");
 }
}
}
```

Question 5

```
/**
*  Gets a command-line argument n (int), and prints an n-by-n damka board.
*/
public class DamkaBoard {
  public static void main(String[] args) {
    int n = Integer.parseInt(args[0]);
    String sOut = "";

    for(int i=1; i<=n; i++){
      sOut = "";

      for(int j=1; j<=n; j++){

        if (i%2!=0){
          sOut =  sOut + "* ";
        } else {
          sOut = sOut + " *";
        }
      }

      System.out.println(sOut);



    }
  }
}
```

Question 6

```
/**
*  Simulates the formation of a family in which the parents decide
*  to have children until they have at least one child of each gender.
*/
public class OneOfEach {
  public static void main (String[] args) {
    boolean Boy = false;
    boolean Girl = false;
    int counter = 0;

    while(!(Boy && Girl)){
      boolean isBoy = Math.random() < 0.5;

      if (isBoy){
        Boy = true;
      }else {
        Girl = true;
      }
      counter++;

      System.out.print(isBoy ? "b" : "g");

      }
    System.out.println();
    System.out.println("You made it... and you now have " + counter + " children.");
    }
  }
```

Question 7

```
/**
*  Computes some statistics about families in which the parents decide
*  to have children until they have at least one child of each gender.
*  The program expects to get one command-line argument: an int value
*  that determines how many families to simulate.
*/
public class OneOfEachStats1 {
  public static void main (String[] args) {
  //initialize variables that will be used
          int T = Integer.parseInt(args[0]);
          int TwoChildren = 0;
          int ThreeChildren = 0;
          int FourOrMore= 0;
          int TotalChildren = 0;
  //for loop which runs T simulations
          for(int i = 0; i<T; i++){
            boolean Boy = false;
            boolean Girl = false;
            int counter = 0;
  //while loop as used previously to check the gender under the assumed probability
and add to count
            while(!(Boy && Girl)) {
               double rnd = Math.random();
               boolean isBoy = rnd < 0.5;

               if (isBoy){
               Boy = true;
               }else {
                  Girl = true;
               }
```

```
                    counter++;

                    }
                TotalChildren += counter;
//checking to see how many children
                if (counter==2){
                    TwoChildren++;
                } else if (counter==3){
                    ThreeChildren++;
                } else{
                    FourOrMore++;
                }
            }
//final calculations and outputs
            double AverageChildren = (double) TotalChildren/T;
            System.out.println("Average: " + AverageChildren + " children to get at least one of each gender.");
            System.out.println("Number of families with 2 children: " + TwoChildren);
            System.out.println("Number of families with 3 children: " + ThreeChildren);
            System.out.println("Number of families with 4 or more children: " + FourOrMore);
//comparing number of children to determine mode
            if(TwoChildren > ThreeChildren && TwoChildren > FourOrMore){
                System.out.println("The most common number of children is 2.");
            } else if (ThreeChildren > TwoChildren && ThreeChildren > FourOrMore){
                System.out.println("The most common number of children is 3.");
            } else{
                System.out.println("The most common number of children is 4 or more.");
            }
        }
 }
```

Question 8

```
import java.util.Random;
/**
*  Computes some statistics about families in which the parents decide
*  to have children until they have at least one child of each gender.
*  The program expects to get two command-line arguments: an int value
*  that determines how many families to simulate, and an int value
*  that serves as the seed of the random numbers generated by the program.
*  Example usage: % java OneOfEachStats 1000 1
*/
public class OneOfEachStats {
  public static void main (String[] args) {
     // Gets the two command-line arguments
     int T = Integer.parseInt(args[0]);
     int seed = Integer.parseInt(args[1]);
     // Initailizes a random numbers generator with the given seed value
     Random generator = new Random(seed);

     //// In the previous version of this program, you used a statement like:
     //// double rnd = Math.random();
     //// Where "rnd" is the variable that stores the generated random value.
     //// In this version of the program, replace this statement with:
     //// double rnd = generator.nextDouble();
     //// This statement will generate a random value in the range [0,1),
     //// just like you had in the previous version, except that the
     //// randomization will be based on the given seed.
     //// This is the only change that you have to do in the program.

     int TwoChildren = 0;
     int ThreeChildren = 0;
     int FourOrMore= 0;
```

```
    int TotalChildren = 0;
//for loop which runs T simulations
    for(int i = 0; i<T; i++){
      boolean Boy = false;
      boolean Girl = false;
      int counter = 0;
//while loop as used previously to check the gender under the assumed probability and
add to count
      while(!(Boy && Girl)) {
        double rnd = generator.nextDouble();
        boolean isBoy = rnd < 0.5;

        if (isBoy){
        Boy = true;
        }else {
          Girl = true;
        }
        counter++;

        }
      TotalChildren += counter;
//checking to see how many children
      if (counter==2){
        TwoChildren++;
      } else if (counter==3){
        ThreeChildren++;
      } else{
        FourOrMore++;
      }

    }
```

```
//final calculations and outputs
    double AverageChildren = (double) TotalChildren/T;
    System.out.println("Average: " + AverageChildren + " children to get at least one of
each gender.");
    System.out.println("Number of families with 2 children: " + TwoChildren);
    System.out.println("Number of families with 3 children: " + ThreeChildren);
    System.out.println("Number of families with 4 or more children: " + FourOrMore);
//comparing number of children to determine mode
    if(TwoChildren > ThreeChildren && TwoChildren > FourOrMore){
      System.out.println("The most common number of children is 2.");
    } else if (ThreeChildren > TwoChildren && ThreeChildren > FourOrMore){
      System.out.println("The most common number of children is 3.");
    } else{
      System.out.println("The most common number of children is 4 or more.");
    }
  }

}
```