

HW02 Code – Noam Adda – ID 209087634

Divisors:

```
public class Divisors {  
    public static void main(String[] args) {  
        // Parse the command line  
        int a = Integer.parseInt(args[0]);  
  
        // Find and print the divisors  
        for (int i = 1; i <= a; i++) {  
            if (a % i == 0) {  
                System.out.println(i);  
            }  
        }  
    }  
}
```

Reverse:

```
public class Reverse {  
    public static void main(String[] args) {  
        // Create the string from the command line argument and get its length  
        String myString = args[0];  
        int length = myString.length();  
  
        // Iterate over all the characters from the end to the beginning and print  
        // each character  
        for (int i = length - 1; i >= 0; i--) {  
            System.out.print(myString.charAt(i));  
        }  
  
        // Define the variable middle, and calculate its position based on the  
        // string length being even or odd  
        int middle;  
        if (length % 2 == 0) {  
            middle = length / 2;  
        } else {  
            middle = (length - 1) / 2;  
        }  
  
        // Print the string middle character  
        System.out.println("\n" + myString.charAt(middle));  
    }  
}
```

InOrder:

```
public class InOrder {  
    public static void main(String[] args) {  
        // Set two variables  
        int generatedNumber = 0;  
        int lastNumber;  
  
        // Iterate when generated number is greater or equal then the last  
        generated number  
        do {  
            lastNumber = generatedNumber;  
            generatedNumber = (int) (Math.random() * 10);  
            // Print only when the generated number is greater or equal than the  
            last generated number  
            if (generatedNumber >= lastNumber) {  
                System.out.print(generatedNumber + " ");  
            }  
        } while (generatedNumber >= lastNumber);  
    }  
}
```

Perfect:

```
public class Perfect {  
    public static void main(String[] args) {  
  
        // Parse the command line number and set it to a variable  
        int num = Integer.parseInt(args[0]);  
  
        // Set the initial sum of divisors  
        int divisorSum = 1;  
  
        // Set the initial text in case of a perfect number  
        String perfectDivisorsText = num + " is a perfect number since " + num +  
        " = 1";  
  
        // Iterate over all numbers from 2 to num and find divisors  
        for (int i = 2; i < num; i++) {  
            if (num % i == 0) {  
                // Sum the divisors and add them to the text  
                divisorSum += i;  
                perfectDivisorsText += (" + " + i);  
            }  
        }  
  
        // Print in case of a perfect number and non-perfect number  
        if (divisorSum == num) {  
            System.out.print(perfectDivisorsText);  
        } else {  
            System.out.print(num + " is not a perfect number");  
        }  
    }  
}
```

DamkaBoard:

```
public class DamkaBoard {  
    public static void main(String[] args) {  
  
        // Parse the command line  
        int n = Integer.parseInt(args[0]);  
  
        // Iterate over rows and columns  
        for (int row = 1; row <= n; row++) {  
  
            // Space before even rows  
            if (row % 2 == 0) {  
                System.out.print(" ");  
            }  
            for (int col = 1; col <= n; col++) {  
                System.out.print("* ");  
            }  
            System.out.println();  
        }  
    }  
}
```

OneOfEach:

```
public class OneOfEach {  
    public static void main(String[] args) {  
  
        // Set the Variables  
        boolean anyBoys = false;  
        boolean anyGirls = false;  
        int numOfChildren = 0;  
        String childrenList = "";  
  
        // Iterate until there's at least one boy and one girl  
        while (!(anyBoys && anyGirls)) {  
  
            // Generate a random number to determine whether the child is a boy  
            or a girl  
            double children = Math.random();  
            if (children < 0.5) {  
                numOfChildren += 1;  
                childrenList += "b ";  
                anyBoys = true;  
            } else {  
                numOfChildren += 1;  
                childrenList += "g ";  
                anyGirls = true;  
            }  
        }  
  
        System.out.println(childrenList);  
        System.out.print("You made it... and you now have " + numOfChildren +  
" children");  
    }  
}
```

}

OneOfEachStats1:

```
public class OneOfEachStats1 {  
    public static void main(String[] args) {  
        // Parse the command line  
        int numOfExperiments = Integer.parseInt(args[0]);  
  
        // Set the variables  
        int numOfAllChildren = 0;  
        int numOfFamiliesWithTwoChildren = 0;  
        int numOfFamiliesWithThreeChildren = 0;  
        int numOfFamiliesWithFourOrMoreChildren = 0;  
  
        // Iterate over number of experiments  
        for (int i = 0; i < numOfExperiments; i++) {  
  
            // Initialize variables per experiment  
            boolean anyBoys = false;  
            boolean anyGirls = false;  
            int numOfChildrenInFamily = 0;  
  
            // Run the experiment until there's one boy and one girl  
            while (!(anyBoys && anyGirls)) {  
                double children = Math.random();  
                if (children < 0.5) {  
                    anyBoys = true;  
                } else {  
                    anyGirls = true;  
                }  
                numOfChildrenInFamily++;  
            }  
        }  
    }  
}
```



```

// Calculate the total number of children
numOfAllChildren += numOfChildrenInFamily;

// Calculate how many families have 2/3/4+ children
if (numOfChildrenInFamily == 2) {
    numOfFamiliesWithTwoChildren++;
} else if (numOfChildrenInFamily == 3) {
    numOfFamiliesWithThreeChildren++;
} else {
    numOfFamiliesWithFourOrMoreChildren++;
}
}

// Calculate the average number of children per family
double avgChildren = (double) numOfAllChildren / (double)
numOfExperiments;

// Print the results of the average and different families
System.out.println("Average: " + avgChildren + " children to get at least
one of each gender.");

System.out.println("Number of families with 2 children: " +
numOfFamiliesWithTwoChildren);

System.out.println("Number of families with 3 children: " +
numOfFamiliesWithThreeChildren);

System.out.println("Number of families with 4 or more children: " +
numOfFamiliesWithFourOrMoreChildren);

// Evaluate and print the result of the most common occurrence
if ((numOfFamiliesWithTwoChildren >=
numOfFamiliesWithThreeChildren) && (numOfFamiliesWithTwoChildren >=
numOfFamiliesWithFourOrMoreChildren)) {
    System.out.println("The most common number of children is 2.");
}

```

```
    } else if ((numOfFamiliesWithThreeChildren >
numOfFamiliesWithTwoChildren) && (numOfFamiliesWithThreeChildren >=
numOfFamiliesWithFourOrMoreChildren)) {
        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.");
    }
}
}
```

OneOfEachStats:

```
public class OneOfEachStats {  
    public static void main(String[] args) {  
        // Gets the two command-line arguments  
        int numOfExperiments = 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);  
  
        // Set the variables  
        int numOfAllChildren = 0;  
        int numOfFamiliesWithTwoChildren = 0;  
        int numOfFamiliesWithThreeChildren = 0;  
        int numOfFamiliesWithFourOrMoreChildren = 0;  
  
        // Iterate over number of experiments  
        for (int i = 0; i < numOfExperiments; i++) {  
  
            // Initialize variables per experiment  
            boolean anyBoys = false;  
            boolean anyGirls = false;  
            int numOfChildrenInFamily = 0;  
  
            // Run the experiment until there's one boy and one girl  
            while (!(anyBoys && anyGirls)) {  
                double children = generator.nextDouble();  
                if (children < 0.5) {  
                    anyBoys = true;  
                } else {
```

```

        anyGirls = true;
    }
    numOfChildrenInFamily++;
}

// Calculate the total number of children
numOfAllChildren += numOfChildrenInFamily;

// Calculate how many families have 2/3/4+ children
if (numOfChildrenInFamily == 2) {
    numOfFamiliesWithTwoChildren++;
} else if (numOfChildrenInFamily == 3) {
    numOfFamiliesWithThreeChildren++;
} else {
    numOfFamiliesWithFourOrMoreChildren++;
}
}

// Calculate the average number of children per family
double avgChildren = (double) numOfAllChildren / (double)
numOfExperiments;

// Print the results of the average and different families
System.out.println("Average: " + avgChildren + " children to get at least
one of each gender.");

System.out.println("Number of families with 2 children: " +
numOfFamiliesWithTwoChildren);

System.out.println("Number of families with 3 children: " +
numOfFamiliesWithThreeChildren);

System.out.println("Number of families with 4 or more children: " +
numOfFamiliesWithFourOrMoreChildren);

```

```
// Evaluate and print the result of the most common occurrence

    if ((numOfFamiliesWithTwoChildren >=
numOfFamiliesWithThreeChildren) && (numOfFamiliesWithTwoChildren >=
numOfFamiliesWithFourOrMoreChildren)) {

        System.out.println("The most common number of children is 2.");

    } else if ((numOfFamiliesWithThreeChildren >
numOfFamiliesWithTwoChildren) && (numOfFamiliesWithThreeChildren >=
numOfFamiliesWithFourOrMoreChildren)) {

        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.");

    }

}

}
```