# Homework 2

1. Divisors

```
public class Divisors {
	public static void main (String[] args) {
		int input_num = Integer.parseInt(args[0]);
		for (int i = 1; i <= input_num; i++){
			if (input_num % i == 0){
				System.out.println(i);
			}
		}



	}
}
```

## 2. Reverse

```
public class Reverse {
  public static void main (String[] args){
        String input_word = args[0];
        int word_length = input_word.length();
        //int middle_character = str.charAt(word_length/2);
        for (int i = (word_length - 1); i>=0; i--){
              System.out.print (input_word.charAt(i));
        }
        System.out.print("\n");
        System.out.println("The middle character is " + input_word.charAt((word_length -
1)/2));


  }
}
```

3. Luck streak

```
public class InOrder {
  public static void main (String[] args) {
                int randmom_num = (int)(Math.random() * 10);
                System.out.println(randmom_num);

                while (true){
                        int new_randmom_num = (int)(Math.random() * 10);
                        if (new_randmom_num >= randmom_num){
                                System.out.println(new_randmom_num);
                                randmom_num = new_randmom_num;
                        }
                        else{
                                break;
                        }

                }




  }
}
```

4. Perfect Numbers

```
public class Perfect {
  public static void main (String[] args) {
        int input_num = Integer.parseInt(args[0]);
        int divisors_sum = 0;
        String perfect_str = ( input_num + " is a perfect number since " + input_num + " = 1");
        for (int i = 2; i < input_num; i++){
              if (input_num % i == 0){
                    divisors_sum += i;
                    perfect_str = (perfect_str + " + " + i);
              }
        }
        if ((divisors_sum + 1) == input_num){
              //perfect_str = (perfect_str + " = " + input_num);
              System.out.println(perfect_str);
        }
        else{
              System.out.println(input_num + " is not a perfect number");
        }

  }
}
```

5. Damka Board

```
public class DamkaBoard {
	public static void main(String[] args) {
		int input_num = Integer.parseInt(args[0]);
		for (int i = 0; i < input_num; i++){
			for (int b = 0; b < input_num; b++){
				System.out.print(((i % 2) == 0) ? "* " : " *" );
			}
		System.out.println();
		}

	}
}
```

6. One of each stats

```
import java.util.Random;
public class OneOfEachStats {
public static String Most_common_fam ( int fam1, int fam2,int fam3){

        int max = fam1;
        String ret_val = "2.";
        if (fam2 > max){
            max = fam2;
            ret_val = "3.";
        }
        if (fam3 > max){
            ret_val = "4 or more.";
        }
        return ret_val;
    }

    public static void main (String[] args) {
        int input_num = 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);
        String mostCommonFam;
        double averageKids = 0;
        boolean  girl = false;
        boolean  boy = false;
        int child_counter = 0;
        int all_childern_counter = 0;
        int familys_of_2 = 0;
        int familys_of_3 = 0;
```

```
int familys_of_4_or_more = 0;
for(int i = 0; i < input_num; i++){
    while (!(boy && girl)){
        if (generator.nextDouble() <= 0.5){
            girl = true;
            //System.out.print("g ");
        }else{
            boy = true;
            //System.out.print("b ");
        }
        child_counter++;
    }
    //System.out.println();
        if (child_counter == 2)
            familys_of_2 ++;
        else if (child_counter == 3)
            familys_of_3 ++;
        else
            familys_of_4_or_more ++;

    //System.out.println(all_childern_counter);
    //System.out.println(child_counter);
    all_childern_counter += child_counter;
    child_counter = 0;
    girl = false;
    boy = false;
}
mostCommonFam = Most_common_fam(familys_of_2, familys_of_3, familys_of_4_or_more);
averageKids = (all_childern_counter/(input_num + 0.0));
System.out.println("Average: " + averageKids + " children to get at least one of each gender.");
```

```
  System.out.println("Number of families with 2 children: " + familys_of_2 );
  System.out.println("Number of families with 3 children: " + familys_of_3 );
  System.out.println("Number of families with 4 or more children: " + familys_of_4_or_more );
  System.out.println("The most common number of children is " + mostCommonFam );

  }
}
```