Hw 2 programs

1. Divisors:

```
class divisors {
    public static void main(String args[])
    {
        int x = Integer.parseInt(args[0]);
        int d = 1;//divisor * k = integer. cant divide in zero.
        while(d < (x/2 + 1))
        {
            if( x % d == 0)
            {
                System.out.println(d + "\n");
            }
            d++;
        }
        System.out.println(x + "\n");
    }
}
```

2. Reversing a string

```java
class Reverse {
    public static void main(String args[])
    {
        String x = args[0];

        for(int i = x.length() - 1; i >= 0; i --)
        {
            System.out.print(x.charAt(i));

        }
        System.out.println("\n" + x.charAt(x.length()/2));
    }
}
```

3. Lucky streak

```
class InOrder {
    public static void main(String args[])
    {
        int max = 10;
        int min = 0;
        int num = (int)(Math.random() * (double)max);;
        int preNum = 0;
        do
        {
            System.out.println(num);
            preNum = num;
            num = (int)(Math.random() * (double)max);
        }while(preNum<=num);
    }
}
```

4. Perfect numbers

```
class Perfect {
    public static void main(String args[])
    {
        int x = Integer.parseInt(args[0]);
        int d = 2;//divisor * k = integer. cant divide in zero, and 1 is known to be a divider.
        int counter = 1;
        String S = args[0] + " is a perfect number since "+args[0]+" = 1";
        while(d < (x/2 + 1))
        {
            if( x % d == 0)
            {
                counter += d;
                S += "+" + String.valueOf(d);
            }
            d++;
        }
        if(counter == x)
        {
            System.out.println(S);
        }
        else
        {
            System.out.println(args[0]+" is not a perfect number");
        }
    }
}
```

5. damka board

```
class DamkaBoard {
    public static void main(String args[])
    {
        int squareSize = Integer.parseInt(args[0]);
        String newLine = "\n";//divisor * k = integer. cant divide in zero, and 1 is known to be a
dividers
        String damkaBoard = "";
        for(int i = 0; i < squareSize; i ++)
        {
            for(int j = 0; j< squareSize; j ++)
            {
                damkaBoard += "* ";
            }
            damkaBoard+= newLine;
            if(i%2 == 0){damkaBoard+= " ";}
        }
        System.out.print(damkaBoard);
    }
}
```

6. One of each

```
class OneOfEach {
    public static void main(String args[])
    {
        int randomSex = (int)(Math.random()*2);// values above 1 - girl, else boy
        char sexToBreak;
        boolean isTwoSexFlag = false;
        int childrenCounter = 1;
        if(randomSex == 1)
        {
            sexToBreak = 'g';
        }
        else
        {
            sexToBreak = 'b';
        }
        do
        {
            childrenCounter++;
            System.out.print(sexToBreak+" ");
            if(randomSex != (int)(Math.random()*2))
            {
                System.out.println((char)(103+98-(int)sexToBreak) + "\n");// ascii substraction to
get b or g when needed.
                isTwoSexFlag = true;//we got another sex
            }
        }while(!isTwoSexFlag);
        System.out.println("You made it... and you now have " +childrenCounter+" children.");
    }

}
```

7. One of each stats 1

```
class OneOfEachStats1 {
    public static void main(String args[])
    {

        int families = Integer.parseInt(args[0]);// values above 1 - girl, else boy
        int twoChildren = 0;
        int threeChildren = 0;
        int fourChildrenPlus = 0;
        int tempo = 0;
        double average = 0;
        String biggestGroup = " 4 or more";
        for(int i = 0; i < families ; i++)
        {
            tempo = familyBuild();//function

            average += tempo;
            switch (tempo){
                case 2:
                    twoChildren ++;
                    break;
                case 3:
                    threeChildren ++;
                    break;
                default:
                    fourChildrenPlus++;
            }

        }
        if(twoChildren>threeChildren && twoChildren> fourChildrenPlus)
        {
            biggestGroup = " 2 ";
        }
        else if(threeChildren> twoChildren && threeChildren > fourChildrenPlus)
        {
            biggestGroup = " 3 ";
        }
        System.out.println("Average: "+ average/families +" children to get at least one of
each gender.\n" +
                "Number of families with 2 children: " + twoChildren +
                "\nNumber of families with 3 children: " + threeChildren +
                "\nNumber of families with 4 or more children: " +fourChildrenPlus +
                "\nThe most common number of children is" + biggestGroup );

    }
    public static int familyBuild()
    {
        int randomSex = (int)(Math.random()*2);// values above 1 - girl, else boy
        boolean isTwoSexFlag = false;
        int childrenCounter = 1;
        do
        {
            childrenCounter++;
            if(randomSex != (int)(Math.random()*2))
            {
                isTwoSexFlag = true;//we got another sex
            }
        }while(!isTwoSexFlag);
        return childrenCounter;
    }
}
```

8. One of each stats

```
import java.util.Random;
class OneOfEachStats {
    public static void main(String args[])
    {

        int families = Integer.parseInt(args[0]);// values above 1 - girl, else boy
        int twoChildren = 0;
        int threeChildren = 0;
        int fourChildrenPlus = 0;
        int tempo = 0;
        double average = 0;
        String biggestGroup = " 4 or more";
        for(int i = 0; i < families ; i++)
        {
            tempo = familyBuild(Integer.parseInt(args[1]));

            average += tempo;
            switch (tempo){
                case 2:
                    twoChildren ++;
                    break;
                case 3:
                    threeChildren ++;
                    break;
                default:
                    fourChildrenPlus++;
            }

        }
        if(twoChildren>threeChildren && twoChildren> fourChildrenPlus)
        {
            biggestGroup = " 2 ";
        }
        else if(threeChildren> twoChildren && threeChildren > fourChildrenPlus)
        {
            biggestGroup = " 3 ";
        }
        System.out.println("Average: "+ average/families +" children to get at least one of
each gender.\n" +
                "Number of families with 2 children: " + twoChildren +
                "\nNumber of families with 3 children: " + threeChildren +
                "\nNumber of families with 4 or more children: " +fourChildrenPlus +
                "\nThe most common number of children is" + biggestGroup );

    }
    public static int familyBuild(int seed)
    {
        Random generator = new Random(seed);
        int randomSex = (int)(generator.nextDouble()*2);// values above 1 - girl, else boy
        boolean isTwoSexFlag = false;
        int childrenCounter = 1;
        do
        {
            childrenCounter++;
            if(randomSex != (int)(generator.nextDouble()*2))
            {
                isTwoSexFlag = true;//we got another sex
            }
        }while(!isTwoSexFlag);
        return childrenCounter;
    }
}
```