

Amir Alexander Fallah

1.

```
public class NumberAboveAverage {
    public static void main(String[] args){
        Scanner keyboard = new Scanner(System.in);
        double sum;
        sum=0;
        double average;
        double[] temperatures = new double[10];
        int count;
        count = 0;
        System.out.print("Enter the temperature for 10 days: ");
        for(int i = 0; i < 10; i++){
            temperatures[i] = keyboard.nextDouble();
            sum += temperatures[i];
        }
        average = sum / 10;
        for(int i = 0; i < temperatures.length; i++){
            if(temperatures[i] > average)count++;
        }
        System.out.print("The average temperature is: " + average);
        System.out.print("The number of temperatures above average: " + count);
    }
}
```

4.

```
public class FlowerCounter {

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int amount;
        int total;
        int currentFlower;
        String flower;
        String[] flowers = {"Petunia", "Pansy", "Rose", "Violet", "Carnation"};
        double[] cost = { 0.50, 0.75, 1.50, 0.50, 0.80};
        System.out.print("The names of the flowers are..");
        for(int i = 0; i < flowers.length; i++){
            System.out.print(" " + flowers[i]);
        }
        System.out.print("Enter a flower name");
        flower = keyboard.next();
        System.out.print("Enter the number of flowers: ");
        amount = keyboard.nextInt();
        for(int j = 0; j < flowers.length; j++ ){
            if(flowers[j].equals(flower)){
                currentFlower = j;
            }
        }

        total = (int) (cost[currentFlower] * amount);
    }
}
```

}

8.

```
public static boolean isStrictlyIncreasing(double[] in)
{
    for(int i = 0; i < in.length - 1; i++){
        if(in[i] >= in[i + 1]){
            return false;
        }
    }
    return true;
}
```

9.

```
public static Character[] removeDuplicates(Character[] in){
    boolean[] result = new boolean[in.length];
    int duplicate = 0;
    boolean temp;
    for(int i = 0; i < in.length; i++){
        temp = true;
        for(int j = 0; j < i; j++){
            if(in[i] == in[j]){
                duplicate++;
                temp = false;
            }
        }
        result[i] = temp;
    }
    Character[] out = new Character[in.length - duplicate];
    int j = 0;
    for(int i = 0; i < result.length; i++){
        if(result[i]){
            out[j] = in[i];
            j++;
        }
    }
    return out;
}
```

}

10.

```
public static int[] remove(int v, int[] in){
    int count;
    count = 0;
    for(int i=0; i < in.length; i++){
        if(in[i]==v)count++;
    }
    int[] out = new int[in.length - count];
    int position;
    position=0;
}
```

```

        for(int j = 0; j < in.length; j++){
            if(in[j] != v)
            {
                out[position] = in[index];
                position++;
            }
        }
        return out;
    }
}

```

20.

```

public static double[][] findFigure(double[][] picture, double threshold)
{
    double[][] newArray = new double[picture.length][picture[0].length];
    double sum = 0.0;
    double average;
    for(int i = 0; i < picture.length; i++)
    {
        for(int j = 0; j < picture[i].length; j++)
            sum+= picture[i][j];
    }
    average = sum / (picture.length * picture[0].length);

    for(int i = 0; i < picture.length; i++)
    {
        for(int j = 0; j < picture[i].length; j++)
        {
            if(picture[i][j] > average * threshold) newArray[i][j] = 1.0;
            else newArray[i][j] = 0.0;
        }
    }
    return newArray;
}

```