Question 1 (12 marks)

- Code a method called sumPosAndNeg that takes an integer array and returns a sum of elements of the array such that:
  - If the first element is positive, sum up only the positive integers but stop if a negative integer is encountered
  - If the first element is negative, sum up only the negative integers but stop if a positive integer is encountered
  - You don't need to deal with an empty array
  - DO NOT REPEAT CODE FOR POSITIVE AND NEGATIVE CASES

```
public class Test {
    public static void main(String[] args) {
        int[]ar1={-1,-6,-2,0,-3,1,5,-1,-6,7}; //-1-6-2-3=-12
        System.out.println(sumPosAndNeg(ar1));
        int[]ar2={2,4,3,1,5,0,1,-1,-3,9,10,2,1}; //2+4+3+1+5+1=16
        System.out.println(sumPosAndNeg(ar2));
    }

    public static int sumPosAndNeg(int[] arr) {
        boolean flag=true;
        int sum=0;
        int i=0;
        while (flag){
            if ((arr[i]>0 && arr[0]<0)||(arr[i]<0 && arr[0]>0)){
                flag=false;
            }
            else{
                sum+=arr[i];
            }
            i++;
        }
        return sum;
    }
}

output:
-12
16
```

Question 2 (14 marks)

Code a method called checkSimilarArrays that takes 2 arrays of integer (ar1 & ar2) and returns an integer value.
The integer indicates how many elements are in ar1 and ar2.

- e.g. an input of [1, 3, 5, 7] and [1, 3, 6, 7] would return 3
- e.g. an input of [4, 2, 9, 8,1] and [0, -3, 9, 8,3] would return 2
- e.g. an input of [] and [2, 3, 4, 5, 6, 7] would return 0
- either or both arrays may come empty or null in which case you must return 0
- An empty array's length is 0

```
public class Test {
    public static void main(String[] args) {
        int[]ar1={};
        int[]ar2={1,5,4,9,11,23,57};
        System.out.println(checkSimilarArrays(ar1,ar2));
        int[]ar3={1,3,5,7};
        int[]ar4={1,3,6,7};
        System.out.println(checkSimilarArrays(ar3,ar4));
        int[]ar5={4,2,9,8,1};
        int[]ar6={0,-3,9,8,3};
        System.out.println(checkSimilarArrays(ar5,ar6));
    }

    public static int checkSimilarArrays (int[]ar1, int[]ar2) {
        int count = 0;
        if (ar1.length == 0 || ar2.length == 0) {
            count = 0;
        }
        else {
            for (int element1 : ar1) {
                for (int element2 : ar2) {
                    if (element1 == element2) {
                        count++;
                    }
                }
            }
        }
        return count;
    }
}

output:
0
3
2
```

```java
public class Test {
    public static void main(String[] args) {
        int[]ar1={};
        int[]ar2={1,5,4,9,11,23,57};
        System.out.println(checkSimilarArrays(ar1,ar2));
        int[]ar3={1,3,5,7};
        int[]ar4={1,3,6,7};
        System.out.println(checkSimilarArrays(ar3,ar4));
        int[]ar5={4,2,9,8,1};
        int[]ar6={0,-3,9,8,3};
        System.out.println(checkSimilarArrays(ar5,ar6));
    }

    public static int checkSimilarArrays(int[]ar1,int[]ar2){
        int count=0;
        for (int i=0;i<ar1.length;i++){
            boolean isFound=false;
            int j=0;
            while (!isFound && j<ar2.length){
                if (ar1[i]==ar2[j]){
                        isFound=true;
                }
                j++;
            }
            if (isFound){
                count++;
            }
        }
        return count;
    }
}
```

output:
0
3
2

Question 3 (12 marks)

- Code a method called calcSeries that takes an integer value n and returns the value of the following series.

$$y = \frac{1}{3} - \frac{4}{6} + \frac{9}{9} - \frac{16}{12} + \cdots \pm \frac{n^2}{n * 3}$$

```
public class Test {
    public static void main(String[] args) {
        System.out.println(calcSeries(3));
    }

    public static double calcSeries(int n) {
        double result=0.0;
        for (int i=1; i<=n; i++){
            double element=((double)(i*i))/(i*3);
            if (i%2!=0){
                result+=element;
            }
            else {
                result-=element;
            }
        }
        return result;
   }
}

output:
0.6666666666666667
```