Homework 4

ArrayOps

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

}

public static int findMissingInt (int [] array) {
    for(int number=0; number<=array.length; number++) {
        boolean isFound = false;
        for(int i=0; i<array.length; i++) {
            if(array[i] == number) {
                isFound = true;
            }
        }
        if(!isFound) {
            return number;
        }



}

return -1;
}

public static int secondMaxValue(int [] array) {
    int max = array[0];
    int max_index=-1;
    for(int i=1; i<array.length; i++) {
        if(array[i] > max) {
            max = array[i];
            max_index = i;
```

```
            }

    }
    array[max_index] = Integer.MIN_VALUE;

     int  max2 = array[0];
    for(int i=1; i<array.length; i++) {
        if(array[i] > max2) {
            max2 = array[i];
        }


}

return max2;

    }

    public static boolean checkElements(int [] a, int [] b) {
        for (int i=0; i<a.length; i++) {
            boolean isFound = false;
            for (int j=0; j<b.length; j++) {
                if(a[i] == b[j]) {
                    isFound = true;

                }
            }

                if(!isFound) {
                    return false;
                }
    }

    return true;
    }



    public static boolean containsTheSameElements(int [] array1,int []
array2) {
        if (checkElements(array1, array2) &&  checkElements(array2,
array1)) {

                return true;
}
else {
```

```
    return false;
}

    }

    public static boolean isSorted(int [] array) {
        boolean increasing = true;
        boolean decreasing = true;

        for (int i = 0; i < array.length - 1; i++) {
            if (array[i] > array[i + 1]) {
                increasing = false;
                break;
            }
        }

        for (int i = 0; i < array.length - 1; i++) {
            if (array[i] < array[i + 1]) {
                decreasing = false;
                break;
            }
        }

        return increasing || decreasing;
    }
}
```

StringOps

```
public class StringOps {

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

```
    }

    public static String capVowelsLowRest (String string) {
        String ans ="";
        for (int i = 0; i < string.length(); i++) {
             char currentChar = string.charAt(i);
             boolean flip =true;
             if  (currentChar==' ' || "AEIOU".indexOf(currentChar)!=-1 )
{
                ans+=currentChar;
            }
             else if (currentChar==' ' && currentChar+1< 'a' ||
currentChar+1< 'a'){
                ans+=(char) (currentChar + 32);
                flip =false;
            } else if ("aeiou".indexOf(currentChar) != -1) {
                ans += (char) (currentChar - 32);
                flip =false;
            }else if (currentChar==' ') {
                ans+=currentChar;
                flip =false;
            } else if (flip ){
            ans+=currentChar;
        }
    }
    return ans;
}

    public static String camelCase (String string) {
            String ans = "";
            int i = 0;

            while (i < string.length()) {
                int firstSpace = string.indexOf(' ', i);
                String helper;
                if (firstSpace == -1) {
                    helper = string.substring(i);
                } else {
                    helper = string.substring(i, firstSpace);
                }
                for (int j = 0; j < helper.length(); j++) {
                    boolean yep =true;
                if (yep) {
                    helper=Makescapitalletters(helper);
                    ans+=helper;
                    break;
```

```
            }
                }

            i += helper.length() + 1;
        }

        return ans =dontMake(ans);
    }

public static int[] allIndexOf (String string, char chr) {
    int index=0;
    int charappears =counterChar(string,chr);
    int [] allIndexOfString=new int[charappears];
    for (int i = 0; i < string.length(); i++) {
        if(string.charAt(i)==chr){
            allIndexOfString[index]=i;
            index++;
        }
    }
    return allIndexOfString;
    }



public static String Makescapitalletters(String string){
        String ans = "";
        boolean flip = true;

        for (int j = 0; j < string.length(); j++) {
            flip = true;
            char currentChar = string.charAt(j);

            if (currentChar >= 'A' && currentChar <= 'Z' && j!=0 ) {
                ans+=((char) (currentChar + 32));
                flip = false;
            } else if (currentChar >= 'a' && currentChar <= 'z') {
                if (j == 0) {
                    ans+=((char) (currentChar - 32));
                    flip = false;
                } else {
                    ans+=(currentChar);
                }
            } else if (flip) {
                ans+=(currentChar);
            }
        }
```

```
            return ans;

}


static public int counterChar(String ans,char letters){
    int counter=0;
    for (int i = 0; i < ans.length(); i++) {
        if(letters==ans.charAt(i)){
            counter++;
        }
    }
    return counter;
}

public static String dontMake(String string){
    String ans="";
    for (int i = 0; i < string.length(); i++) {
    if (string.charAt(0)>= 'A' && string.charAt(0) <= 'Z' && i==0){
        ans+=(char) (string.charAt(i)+32);
    }
    else
    ans+=string.charAt(i);

}
return ans;
}
}
```