HW4

ArrayOps.java

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

    }

    public static int findMissingInt (int [] array) {
        int n = array.length;
        if (n == 1) {
            return 1;
        }

         for (int i = 0; i < n; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
        for (int i = 0; i < n; i++){
            int y = array[i];
            if (y != i){
                return i;
            }
        }
        return n;
    }

    public static int secondMaxValue(int [] array) {
        int n = array.length;

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }
        return array[(n - 2)];
    }

    public static boolean containsTheSameElements(int [] array1,int [] array2) {
        boolean SameElements = true;
```

```java
        int n1 = array1.length;
        int n2 = array2.length;
        for (int i = 0; i < n1; i++){
            for (int j = 0; j < n2; j++){
                if (array1[i] == array2[j]) {
                    break;
                } else if (j == (n2 - 1)){
                    SameElements = false;
                }
            }
        }
        return SameElements;
    }

    public static boolean isSorted(int [] array) {
        boolean Sorted = true;
        int n = array.length;
        int[] ord = new int[n];
        for (int i = 0; i < n; i++){
            ord[i] = array[i];
        }

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (ord[j] > ord[j + 1]) {
                    int temp = ord[j];
                    ord[j] = ord[j + 1];
                    ord[j + 1] = temp;
                }
            }
        }

        int[] rev = new int[n];
        for (int i = 0; i < n; i++) {
            rev[i] = ord[n - i - 1];
            }

        for (int i = 0; i < n; i++) {
            if (array[i] != rev[i] && array[i] != ord[i]) {
                Sorted = false;
            }
        }
        return Sorted;
    }
}
```

StringsOps.java :

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

    }

    public static String capVowelsLowRest (String string) {
        String x = "";
        int n = string.length();
        for (int i = 0; i < n; i++){
            int y = (char)(string.charAt(i));
            if (string.charAt(i) == ' '){
                x = x + ' ';
            } else if (y == 97 || y == 65){
                x = x + "A";
            } else if (y == 101 || y == 69){
                x = x + "E";
            } else if (y == 105 || y == 73){
                x = x + "I";
            } else if (y == 111 || y == 79){
                x = x + "O";
            } else if (y == 117 || y == 86){
                x = x + "U";
            } else if (y > 65 && y < 69){
                x = x + ((char)(y + 32));
            } else if (y > 69 && y < 73){
                x = x + ((char)(y + 32));
            } else if (y > 73 && y < 79){
                x = x + ((char)(y + 32));
            } else if (y > 79 && y < 85){
                x = x + ((char)(y + 32));
            } else if (y > 85 && y < 91){
                x = x + ((char)(y + 32));
            } else {
                x = x + string.charAt(i);
            }
        }
        return x;
    }

    public static String camelCase (String string) {
        String x = "";
        int n = string.length();
        for (int i = 0; i < n; i++){
            int y = (char)(string.charAt(i));
```

```
            if (string.charAt(i) == ' '){

                i++;

                while (i < n - 1 && string.charAt(i) == ' ') {
                    i++;
                }

                if (i == n){

                    return x;

                } else if (i == 1){
                    if (((char)(string.charAt(i))) > 64 &&
((char)(string.charAt(i)) < 91)){
                        x = x + ((char)(string.charAt(i) + 32));
                    } else {
                        x = x + (string.charAt(i));
                    }
                } else {
                    if (((char)(string.charAt(i))) > 64 &&
((char)(string.charAt(i)) < 91)) {
                            x = x + (string.charAt(i));
                    } else {
                        x = x + ((char)(string.charAt(i) - 32));
                    }

                }

            } else if (y >= 65 && y <= 90){

                x = x + ((char)(y + 32));

            } else {

                x = x + string.charAt(i);

            }
        }
        return x;
    }

    public static int[] allIndexOf (String string, char chr) {
        int x = 0;
        int n = string.length();
        for (int i = 0; i < n; i++){
            if (string.charAt(i) == chr){
                x++;
            }
        }
        int [] z = new int [x];
```

```
        int j = 0;
        for (int i = 0; i < n; i++){
            if (string.charAt(i) == chr){
                z[j] = i;
                j++;
            }
        }

        return z;
    }

}
```