```java
public class ArrayOps {
    public static void main(String[] args) {
        int[] array = {1, 3, 2};

        System.out.println(isSorted(array));

    }

    public static int findMissingInt (int [] array) {
        int N = array.length;
        int missingNum = 0;
        int[] temp = new int[N+1];
        for (int i = 0; i < N; i++) {
            temp[array[i]]++;
            }
            for (int j = 0; j < N+1; j++) {
             if (temp[j] == 0) {
                missingNum = j;
             }
            }
        return missingNum;
    }

    public static int secondMaxValue(int [] array) {
        int max = Integer.MIN_VALUE;
        int secMax = Integer.MIN_VALUE;

        for (int i = 0; i < array.length; i++) {
            if (array[i] > max) {
                secMax = max;
                max = array[i];
            }else if( array[i] != max || array[i] > secMax) {
                secMax = array[i];
            }
        }
        return secMax;
    }
```

```
    public static boolean containsTheSameElements(int [] array1,int [] array2) {
        for (int i = 0; i < array1.length; i++) {
            for (int j = 0; j < array2.length; j++) {
             if (array1[i] == array2[j]){
             break;
             }
             else if (j == array2.length - 1) {
             return false;
             }
        }
    }
    for (int i = 0; i < array2.length; i++) {
        for (int j = 0; j < array1.length; j++) {
         if (array2[i] == array1[j]){
         break;
         }
         else if (j == array1.length - 1) {
         return false;
         }
    }
}
        return true;
    }
```

```java
    public static boolean isSorted(int [] array) {
        int num1 = array[0];
        int num2 = array[1];
        boolean decrease = (num1 <= num2);
        boolean Increase = (num1 >= num2);

        if (decrease){
            for (int i = 1; i < array.length - 1; i++){
                num1 = num2;
                num2 = array[i+1];
                if (num1 >= num2){
                    return false;
                }
            }
        }
        else if (Increase){
            for (int i = 1; i < array.length - 1; i++){
                num1 = num2;
                num2 = array[i+1];
                if (num1 <= num2){
                    return false;
                }
            }
        }
        return true;
    }

}
public class StringOps {
    public static void main(String[] args) {
        String string = "MMMM";
        char chr = 'M';
        int[] result = allIndexOf(string, chr);

        for (int i = 0; i < result.length; i++) {
            System.out.print(result[i]);
            if (i < result.length - 1) {
                System.out.print(" ");
            }
    }
}
```

```
   public static String capVowelsLowRest (String string) {
      String ans = "";
      for (int i = 0; i < string.length(); i++) {
         char ch = string.charAt(i);
         if ((ch >= 65) && (ch <= 90)){   //if big letter
            if (ch == 65 || ch == 69 || ch == 73 || ch == 79 || ch == 85){ //if aeiou
               ans = ans + ch; //add as is
            }
            else{
               ans = ans + ((char)(ch + 32)); //add as lowercase
            }
         }
         else if ((ch >= 97) && (ch <= 122) || ch == 32){
         if (ch == 97 || ch == 101 || ch == 105 || ch == 111 || ch == 117)
{                        // if small letter or space
               ans = ans + ((char)(ch - 32)); //add as uppercase
         }
         else {                        // if its any other char
            ans = ans + ch;               // add as is
         }
      }
   }
      return ans;
}
```

```java
public static String camelCase (String string) {
        int i = 0;
        int length = string.length();

            while (i < length && string.charAt(i) == ' ' ) {  //if the string starts with
spaces we will skip them
                i++;
            }

        String ans = "";
            while (i < length) {

        char ch = string.charAt(i);

                if (ch == ' '){

             while (i < length && string.charAt(i) == ' ' ) {  //if there are spaces we will
skip them
                        i++;
                }

                if (i < length){
                    ch = string.charAt(i);
                    if (ch >='a' && ch <= 'z'){
                        ans += (char)(ch - 32);
                    }else {
                        ans += ch;
                    }

                }
                i++;
            }
             else if (ch >= 'A' && ch <= 'Z'){
                ans += (char)(ch + 32);
                i++;
             }
             else {
                ans += ch;
                i++;
             }
        }
        return ans;
 }
```

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

        int[] ans = new int[count];
        int currentIndex = 0;

        for (int i = 0; i < string.length(); i++) {
            if (string.charAt(i) == chr) {
                ans[currentIndex] = i;
                currentIndex++;
            }
        }

        return ans;
    }
}
```