

```

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

    }

    public static int findMissingInt (int [] array) {

        boolean exist=false;

        for(int j=0; j<=array.length; j++){
            for(int i=0; i<array.length; i++){
                if(array[i]==j) exist=true;
                if(exist==false && i==array.length-1) return
j;
            }
            exist=false;
        }
        return -1;
    }

    public static int secondMaxValue(int [] array) {
        int max=0;
        int max2=0;
        int count=0;
        for(int i=0; i<array.length; i++){
            if(array[i]>max) max=array[i];
        }
        for(int j=0; j<array.length; j++){
            if(max==array[j]) count++;
        }
        if(count>1) return max;
        for(int r=0; r<array.length; r++){
            if(array[r]>max2 && array[r]!=max) max2=array[r];
        }
        return max2;
    }

    public static boolean containsTheSameElements(int []
array1,int [] array2) {
        boolean exist=false;

        for(int j=0; j<array1.length; j++){
            for(int i=0; i<array2.length; i++){
                if(array1[j]==array2[i]) exist=true;
                if(exist==false && i==array2.length-1) return
false;
            }
            exist=false;
        }
    }
}

```

```

    }

    for(int j=0; j<array2.length; j++){
        for(int i=0; i<array1.length; i++){
            if(array2[j]==array1[i]) exist=true;
            if(exist==false && i==array1.length-1) return
false;
        }
        exist=false;
    }

    return true;
}

public static boolean isSorted(int [] array) {
    if(array[0] > array[array.length-1]){
        for(int i=0; i<array.length-1; i++){
            if(array[i]<array[i+1]) return false;
        }
    }
    else if(array[0] < array[array.length-1]){
        for(int i=0; i<array.length-1; i++){
            if(array[i]>array[i+1]) return false;
        }
    }
    return true;
}
}

```

```

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

        public static String capVowelsLowRest (String string) {
            String help = "";
            for(int i=0; i<string.length(); i++){
                if(string.charAt(i)=='a' || string.charAt(i)=='A')
help+='A';
                else if(string.charAt(i)=='e' ||
string.charAt(i)=='E') help+='E';
                else if(string.charAt(i)=='i' ||
string.charAt(i)=='I') help+='I';
                else if(string.charAt(i)=='o' ||
string.charAt(i)=='O') help+='O';
                else if(string.charAt(i)=='u' ||
string.charAt(i)=='U') help+='U';
                else if(string.charAt(i)==' ') help+=' ';
                else if (string.charAt(i)<=90 &&
string.charAt(i)>=65) help+=(char)(string.charAt(i)+32);
                else help+=string.charAt(i);
            }
            return help;
        }

        public static String camelCase (String string) {
            String help = "";
            boolean afterspace = false;
            boolean firstspace = false;

            for(int i=0; i<string.length(); i++){
                if(string.charAt(i)==' '){
                    if(i==0) firstspace = true;
                    afterspace=true;
                }
                else if(i==0){
                    if (string.charAt(i)<=90 &&
string.charAt(i)>=65) help+=(char)(string.charAt(i)+32) ;
                    else help+=string.charAt(i);
                }
                else if(afterspace==true && string.charAt(i)!='
'){
                    if(firstspace==false){
                        if (string.charAt(i)<=90 &&
string.charAt(i)>=65) help+=string.charAt(i);
                        else help+=(char)(string.charAt(i)-32);
                        afterspace=false;
                    }
                }
            }
        }
    }
}

```

```

        else if (firstspace=true){
            if (string.charAt(i)<=90 &&
string.charAt(i)>=65) help+=(char)(string.charAt(i)+32) ;
            else help+=string.charAt(i);
            firstspace=false;
            afterspace=false;
        }
    }
    else if (string.charAt(i)<=90 &&
string.charAt(i)>=65) help+=(char)(string.charAt(i)+32);
    else help+=string.charAt(i);
}
return help;
}

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 [] end =new int[count];
    boolean isin=false;
    int r=0;

    for(int j=0; j<end.length; j++){
        while (isin==false) {
            if (string.charAt(r)==chr){
                end[j]=r;
                isin=true;
            }
            r++;
        }
        isin=false;
    }

    return end;
}
}

```