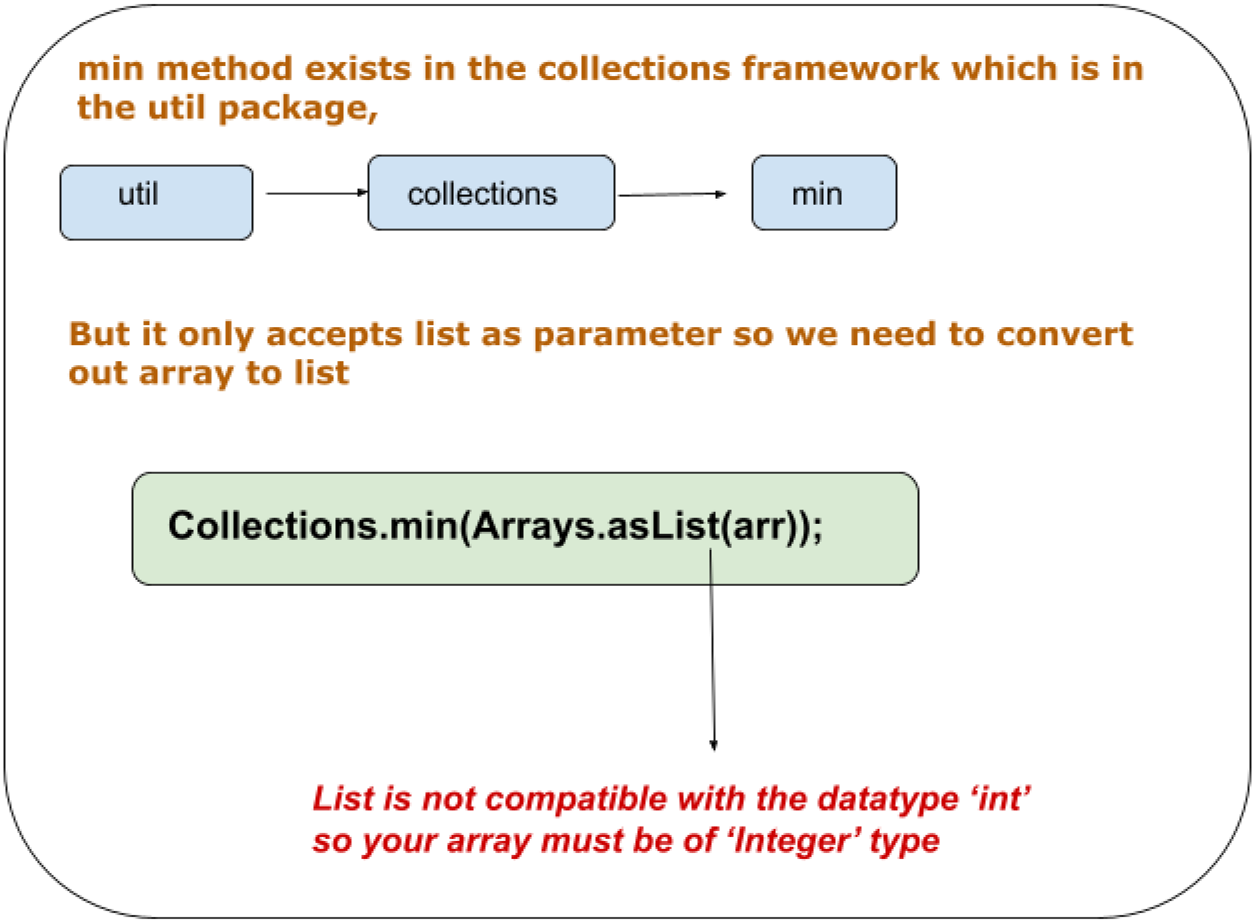
Array Operations

Minimum Operation

It is used to find the minimum(smallest either numerically or alphabetically) value in the given array



min method exists in the collections framework which is in the util package,

util → collections → min

But it only accepts list as parameter so we need to convert out array to list

```
Collections.min(Arrays.asList(arr));
```

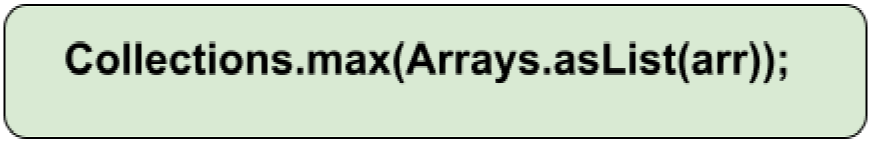
List is not compatible with the datatype 'int' so your array must be of 'Integer' type

```
// int[] arr = {1,2,3,4,5};   -> can give error
       //                        because list is not
       //                        compatible with int

       Integer[] arr = { 1, 2, 3, 4, 5, 6, 0 };
       int min = Collections.min(Arrays.asList(arr));

       System.out.println(min);
```

```
String[] str = {"AA","AB","AC"};
       String str_min = Collections.min(Arrays.asList(str));
       System.out.println(str_min);
       //output:- "AA"
```

Maximum Operation

Finding the largest element in the array either numerically or alphabetically.



```
Collections.max(Arrays.asList(arr));
```

Q. sum of Max and Min

Problem Statement

Suggest Edit

You are given an array “ARR” of size N. Your task is to find out the sum of maximum and minimum elements in the array.

Follow Up:

Can you do the above task in a minimum number of comparisons?

Input Format:

The first line of input contains a single integer T, representing the number of test cases.
Then the T test cases follow.

The first line of each test case contains a single integer N representing the size of the array 'ARR'.

The second line of each test case contains N space separated integers representing the elements of the array “arr”.

Output Format:

For each test case, print the sum of the maximum and minimum element of the array 'ARR'.

Note:

You do not need to print anything. It has already been taken care of. Just implement the given function.

Constraints:

1 <= T <= 10
1 <= N <= 10^5
-10^9 <= ARR[i] <= 10^9

Time limit: 1 second

//code

```
 import java.util.*;
 public class Solution {
   public static int sumOfMaxMin(int[] arr, int n) {
       // Write your code here.

       Integer[] ar = new Integer[n];

       for(int i = 0; i < n; i++){
           ar[i] = Integer.valueOf(arr[i]);
       }

       int max = Collections.max(Arrays.asList(ar));
       int min = Collections.min(Arrays.asList(ar));

       return max+min;

   }
 }
```

Sort Operation

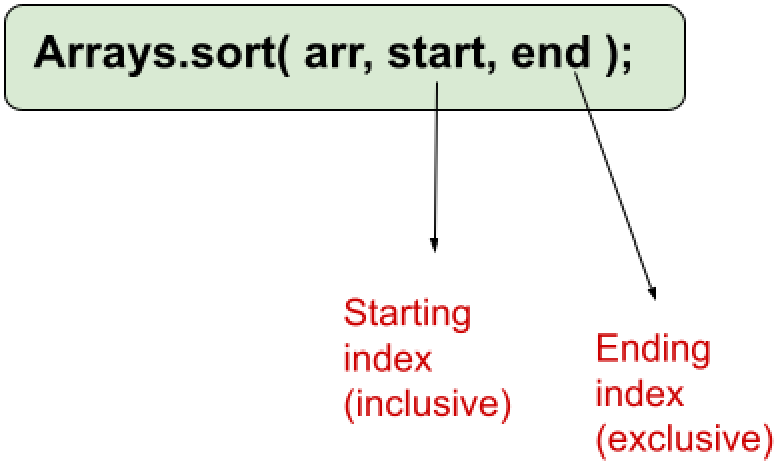
Elements are ordered either in ascending order or descending order.

Arrays.sort(arr);

```
    int[] arr = {11,-2,-3,4,5,6,111};
    Arrays.sort(arr);
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i]+" ");
    }

    //output:-  -3 -2 4 5 6 11 111
```

```
String[] str = {"AA","TDS","ZLX","M","YU","AB","AZ"};
    Arrays.sort(str);
    for (int i = 0; i < str.length; i++) {
        System.out.print(str[i]+" ");
    }

    //output:- AA AB AZ M TDS YU ZLX
```

Sorting specific part of array



Arrays.sort(arr, start, end);

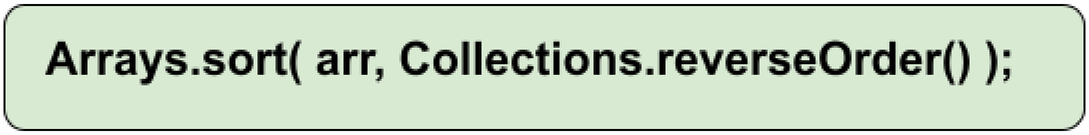
Starting index (inclusive)

Ending index (exclusive)

```
int[] arr = {11,-2,-3,4,5,6,111};
       //Arrays.sort(arr);

       Arrays.sort(arr,3,6);

       for (int i = 0; i < arr.length; i++) {
           System.out.print(arr[i]+" ");
       }

       //output:- -3 -2 4 5 6 11 111
```

```
String[] str = {"AA","TDS","ZLX","M","YU","AB","AZ"};
       //Arrays.sort(str);
       Arrays.sort(str,3,6);
       for (int i = 0; i < str.length; i++) {
           System.out.print(str[i]+" ");
       }
       //output:- AA TDS ZLX AB M YU AZ
```

Reverse order function

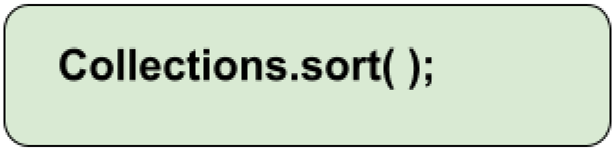


```
Arrays.sort( arr, Collections.reverseOrder() );
```

```
Integer[] arr = {11,-2,-3,4,5,6,111};
       Arrays.sort(arr,Collections.reverseOrder());
       //for this int is not allowed we need Integer
       for (int i = 0; i < arr.length; i++) {
           System.out.print(arr[i]+" ");
       }
       //output:- 111 11 6 5 4 -2 -3
```

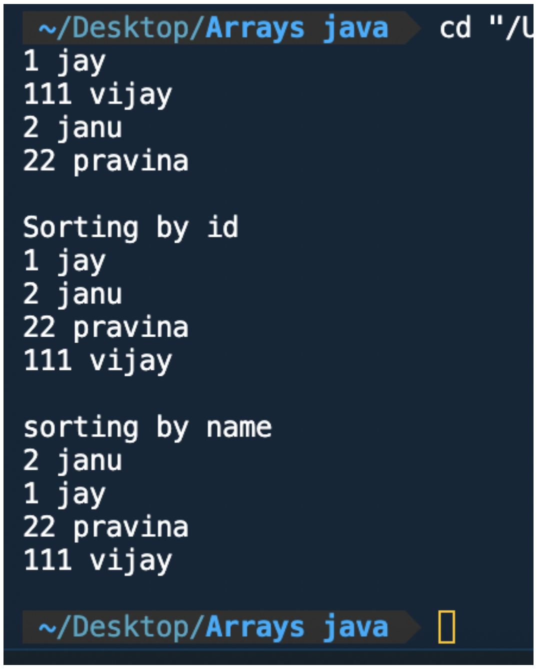
📝 ***Note:***

If we want to sort array of user defined classes then we need to use



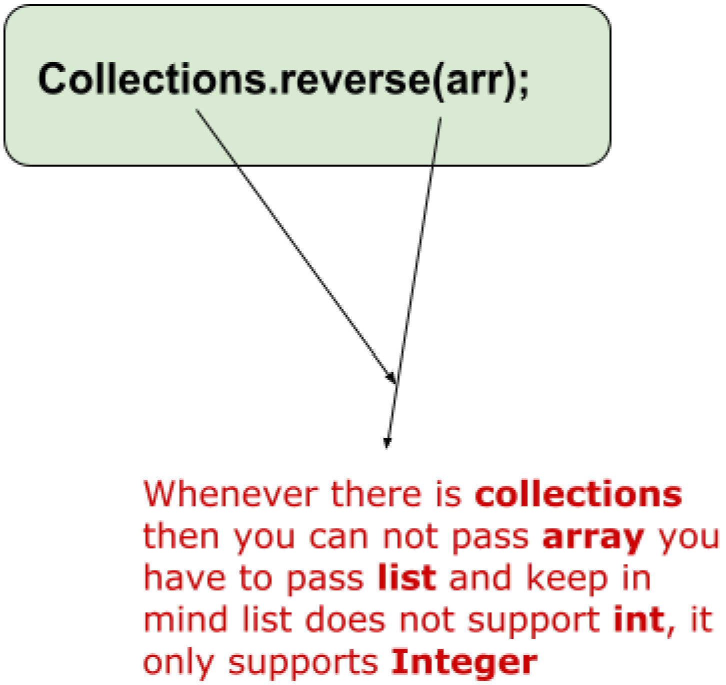
```
Collections.sort( );
```

```java
import java.util.*;

//custom class
class Person{
   int id;
   String name;

   public Person(int id, String name){
       this.id = id;
       this.name = name;
   }
   public String displayPerson(){
       return id+" "+name;
   }
}

//comparator interface used
class SortById implements Comparator<Person>{
   @Override
   public int compare(Person o1, Person o2) {
       return o1.id - o2.id;
   }
}

//comparator interface used
class SortByName implements Comparator<Person>{
   @Override
   public int compare(Person o1, Person o2) {
       return o1.name.compareTo(o2.name);
   }
}

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

       ArrayList<Person> list = new ArrayList<>();
       list.add(new Person(1, "jay"));
       list.add(new Person(111, "vijay"));
       list.add(new Person(2, "janu"));
       list.add(new Person(22, "pravina"));

       //normal order or list
       for (Person pr : list) {
           System.out.println(pr.displayPerson());
       }
       System.out.println();

       //sorting by id
       Collections.sort(list, new SortById());
       System.out.println("Sorting by id");
       for (Person pr : list) {
           System.out.println(pr.displayPerson());
       }
       System.out.println();

       //sorting by name
       Collections.sort(list, new SortByName());
       System.out.println("sorting by name");
       for (Person pr : list) {
           System.out.println(pr.displayPerson());
       }
       System.out.println();

   }
}
```



```
 ~/Desktop/Arrays java  cd "/U
1 jay
111 vijay
2 janu
22 pravina

Sorting by id
1 jay
2 janu
22 pravina
111 vijay

sorting by name
2 janu
1 jay
22 pravina
111 vijay

 ~/Desktop/Arrays java
```

Reverse Operation



Collections.reverse(arr);

Whenever there is **collections** then you can not pass **array** you have to pass **list** and keep in mind list does not support **int**, it only supports **Integer**

Q. reverse the array

```
import java.util.*;
class Solution {

      public static void main(String args[]) {

            // Write code here

        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        ArrayList<Integer> arr = new ArrayList<>();
        for(int i = 0; i < n; i++){
            arr.add(sc.nextInt());
```

```
    }

    Collections.reverse(arr);

    for(Integer i : arr){
      System.out.print(i+" ");
    }

   }
}
```

Q. Reverse words in a string

```
import java.util.*;
public class Solution
{
   public static String reverseString(String str)
   {
      //Write your code here

    List<String> list = new ArrayList<String>();
    String t = "";

    for(int i = 0; i <= str.length(); i++){

      if(i==str.length()){
        if(t!=""){
          list.add(t);
        }

      }else{

        if(str.charAt(i)==' '){
          if(t!=""){
            list.add(t);
            t="";
          }
        }
```

```java
                else{
                    t+=str.charAt(i);
                }

            }
        }
        Collections.reverse(list);

        String ans = "";

            for(int i = 0; i < list.size();i++){
                ans = ans + list.get(i);
                ans = ans + " ";

            }

        return ans;

      }
}
```