A mountain climbing club maintains a record of the climbs that its members have made. Information about a climb includes the name of the mountain peak and the amount of time it took to reach the top. The information is contained in the ClimbInfo class as declared below.

```
public class ClimbInfo
{
    /** Creates a ClimbInfo object with name peakName and time climbTime.
    * @param peakName the name of the mountain peak
    * @param climbTime the number of minutes taken to complete the climb
    */
    public ClimbInfo(String peakName, int climbTime)
    {        /* implementation not shown */ }

    /** @return the name of the mountain peak
    */
    public String getName()
    {        /* implementation not shown */ }

    /** @return the number of minutes taken to complete the climb
    */
    public int getTime()
    {        /* implementation not shown */ }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

The ClimbingClub class maintains a list of the climbs made by members of the club. The declaration of the ClimbingClub class is shown below. You will write two different implementations of the addClimb method. You will also answer two questions about an implementation of the distinctPeakNames method.

```
public class ClimbingClub
  / * * The list of climbs completed by members of the club.
   * Guaranteed not to be null. Contains only non-null references.
  private List<ClimbInfo> climbList;
  /** Creates a new ClimbingClub object. */
  public ClimbingClub()
  { climbList = new ArrayList<ClimbInfo>(); }
  /** Adds a new climb with name peakName and time climbTime to the list of climbs.
    * @param peakName the name of the mountain peak climbed
       @param climbTime the number of minutes taken to complete the climb
  public void addClimb(String peakName, int climbTime)
  { /* to be implemented in part (a) with ClimbInfo objects in the order they were added */
      /* to be implemented in part (b) with ClimbInfo objects in alphabetical order by name */
  }
  /** @return the number of distinct names in the list of climbs */
  public int distinctPeakNames()
  { /* implementation shown in part (c) */ }
  // There may be instance variables, constructors, and methods that are not shown.
```

(a) Write an implementation of the ClimbingClub method addClimb that stores the ClimbInfo objects in the order they were added. This implementation of addClimb should create a new ClimbInfo object with the given name and time. It appends a reference to that object to the end of climbList. For example, consider the following code segment.

```
ClimbingClub hikerClub = new ClimbingClub();
hikerClub.addClimb("Monadnock", 274);
hikerClub.addClimb("Whiteface", 301);
hikerClub.addClimb("Algonquin", 225);
hikerClub.addClimb("Monadnock", 344);
```

When the code segment has completed executing, the instance variable climbList would contain the following entries.

Peak Name Climb Time

"Monadnock"	"Whiteface"	"Algonquin"	"Monadnock"
274	301	225	344

```
Information repeated from the beginning of the question

public class ClimbInfo

public ClimbInfo(String peakName, int climbTime)

public String getName()

public int getTime()

public class ClimbInfo> climbList

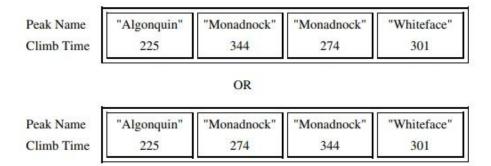
public void addClimb(String peakName, int climbTime)

public int distinctPeakNames()
```

(b) Write an implementation of the ClimbingClub method addClimb that stores the elements of climbList in alphabetical order by name (as determined by the compareTo method of the String class). This implementation of addClimb should create a new ClimbInfo object with the given name and time and then insert the object into the appropriate position in climbList. Entries that have the same name will be grouped together and can appear in any order within the group. For example, consider the following code segment.

```
ClimbingClub hikerClub = new ClimbingClub();
hikerClub.addClimb("Monadnock", 274);
hikerClub.addClimb("Whiteface", 301);
hikerClub.addClimb("Algonquin", 225);
hikerClub.addClimb("Monadnock", 344);
```

When the code segment has completed execution, the instance variable climbList would contain the following entries in either of the orders shown below.



You may assume that climbList is in alphabetical order by name when the method is called. When the method has completed execution, climbList should still be in alphabetical order by name.

```
Information repeated from the beginning of the question

<u>public class ClimbInfo</u>

public ClimbInfo(String peakName, int climbTime)

public String getName()

public int getTime()

<u>public class ClimbingClub</u>

private List<ClimbInfo> climbList

public void addClimb(String peakName, int climbTime)

public int distinctPeakNames()
```

(c) The ClimbingClub method distinctPeakNames is intended to return the number of different names in climbList. For example, after the following code segment has completed execution, the value of the variable numNames would be 3.

```
ClimbingClub hikerClub = new ClimbingClub();
hikerClub.addClimb("Monadnock", 274);
hikerClub.addClimb("Whiteface", 301);
hikerClub.addClimb("Algonquin", 225);
hikerClub.addClimb("Monadnock", 344);
int numNames = hikerClub.distinctPeakNames();
```

Consider the following implementation of method distinctPeakNames.

```
/** @return the number of distinct names in the list of climbs */
public int distinctPeakNames()
  if (climbList.size() == 0)
  {
    return 0;
  ClimbInfo currInfo = climbList.get(0);
  String prevName = currInfo.getName();
  String currName = null;
  int numNames = 1;
  for (int k = 1; k < climbList.size(); k++)
    currInfo = climbList.get(k);
    currName = currInfo.getName();
    if (prevName.compareTo(currName) != 0)
       numNames++;
       prevName = currName;
  return numNames;
```

Ass	ume that addClimb works as specified, reg	gardless of what you wrote in parts (a) and (b).
(i)		t PeakNames method work as intended when the objects in the order they were added as described in
	Circle one of the answers below.	
	YES	NO

(ii) Does this implementation of the distinctPeakNames method work as intended when the addClimb method stores the ClimbInfo objects in alphabetical order by name as described in part (b)?

Circle one of the answers below.

YES NO