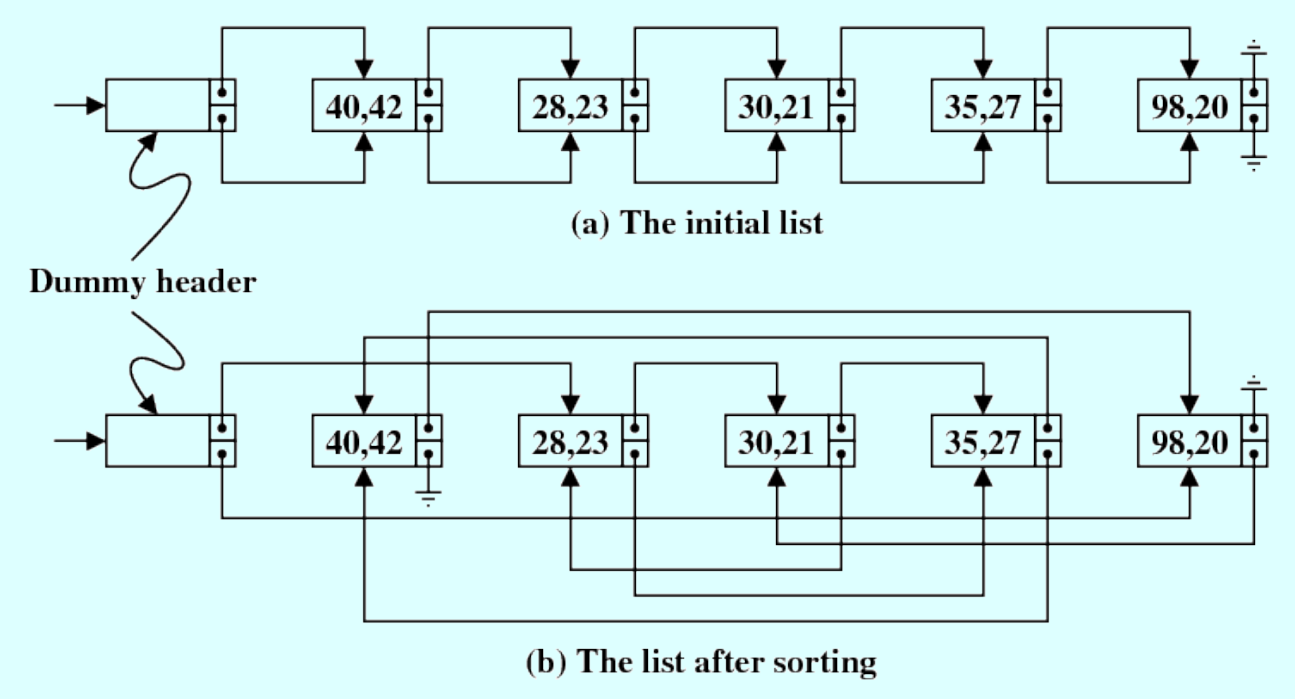
Sorting a list of points:

We are given a linked list of points in the X-Y plane. We want to sort the list simultaneously with respect to the X-coordinates and with respect to the Y-coordinates. To achieve this goal, we maintain two pointers in each node. One set of these pointers (nextX) is used to sort the list with respect to the X-values, the other (nextY) with respect to the Y-values. The following figure demonstrates this.



Design a data structure for implementing the linked list of points.

CreateList(n): Generate a random list of **n** points with integer coordinates. At this point, you do not worry about sorting of the elements, but insert every new element in the list with respect to both the nextx and nexty pointers.

PrintList (list L, int flag): This function prints the list pointed to by the pointer L. Now, each node has two pointers. The flag indicates whether the list is to be traversed along the nextx pointers, or along the nexty pointers.

SortList (list L): Write two functions with the following prototypes:

```
void SortX ( list L );   void SortY ( list L );
```

The first function is meant for sorting the list headed by L with respect to the x-values, and the second with respect to the y-values. You are not supposed to separate the x and y coordinates of a point. So you must adjust the relevant pointers in the nodes in order to effect two independent sorting of the same list.