```
SecondSort( List songsByArtist )
{
// the parameter is a list where there are groups of songs sorted by artist, but the song list within each artist
group is not sorted

int i = 0;
int j = 0;
while( j < n )
{
   artist = songsByArtist[i];
   j = i;

   while (artist == songsByArtist[j]
     && j < n)
   {
         j = j + 1;
   }

   artistSubArray = songsByArtist[i to j-1]

   InsertionSort(artistSubArray);
   i = j;
}
```

It took me a little while to realize what was happening in this function. I believe that is what is happening is that each element is sorted by 1: starting with the base case, making sure that each element up to j is by the same artist. In the base case there will only be one element. 2: Then an array of those songs is created from the input, songsByArtist. 3. Each of those arrays is progressively run through the original sorting algorithm. It is done one element at a time. By the time that the array hits a song not by the same artist or the end of the list, each element will be sorted. As long as the input is a set of songs the list will be sorted by artist. Since the inner while loop sorts out songs that are not of the same artist it does not have to be the same.

My orignal sorting algorithm was very basic and was the InsertionSort mentioned above.
insertion sort (givenlisty{

List = Artist/music list

For (iterate over the artist list)

For(iterate and make sure everything is in order)
This will make each position in the for loop in order

}