 Declare an array of at least 10 Music objects. For each, you will need a song title, year, and artist name. At least one year needs to have multiple entries in the array. Same with one of the artists. Of course, be sure to use school-appropriate songs.  
**For example:** Livin' on a Prayer, 1986, Bon Jovi

 Design a static method that traverses through the array and prints each element.

 Since the data will need to be sorted prior to conducting a binary search, three static methods to do so need to be created.

1. Write static methods to sort by title, year, and artist. You may use the insertion, selection, or merge sort, but not a bubble sort.
2. Name and document the methods to clearly indicate the type of sort and the values being sorted.
3. Utilize print debugging statements to ensure the sorts worked. Be sure to comment these out prior to submitting your work.

 Create the following static methods in the tester class. Utilize the binary search algorithm. Each method will take two arguments: the array and the value to find.

1. a method that searches the array for a particular song title
2. a method that searches the array for year released. The output should list all songs found from that year
3. a method that searches the array for the name of the artist. The output should list all songs performed by that artist
4. methods to assist with printing all matches after a binary search has found a match. Model your code after the linearPrint method sample

**Easy to understand version**

* Declare an array of 10 songs including artist, year, and title
* Have a static method that will print out all the songs
* Create a methods that will sort the songs based on title, year, and author
  + Clearly label how its being sorted
  + Feel free to make debug statements when needed
* Create the following methods in the binary search algo (use provided examples)
  + Searches the list for a song
  + A method that searches by year released (include multiples if needed)
  + A method that searches by artist (include multiples if needed)
  + A method that prints all matches once item is found (use LinearPrint in example as a base)