Q1:

consider a list of persons(names)and their phone numbers .you have to create a contact list (names and the phone numbers).You did not need to add more contacts to the list but frequently search for contacts based on their names and in the list.Among the list data structures you have studied ,recommend one data structure that is suitable for the scenario and also state your arguments to justify your selection.provide definition of class required for its implementation .YOu do not need to implement the fuction such as insert (),delete()and search()etc.just list the class and their data members

**ANS:** From my naive point of view, at first I thought of implementing it using a hash table where two distinct keys, i.e name and phone number points to one single record, but doing something like is a tough task for an average lazy programming student like me, moreover this may not be the optimised way of performing the operations mentioned above. So this time I took some horlics and got a better idea.. “TRIEs” , Why?? Here’s the reason..

1. For contact list I may have lot of prefix related queries and TRIE works best over such queries.
2. A poor hash function may map my contacts “Girlfriend” and “Girlfriend-2” to same value making it difficult for me to decide which one to call [that is my pure imagination BTW] which is not the case with tries.
3. And obviously how can I forget the most favourite factor of a programmer “Running Time”, trie comes with a predictable lookup time and hash tables…did I mentioned the word “predictable” ?

Hope this has helped you, Happy reading !!!! :-)