Question 1:

* For linked list:

I have made a template type linked list so that we can deal with both characters and integers.

All the functions are written in linked list like insert (add ()), delete node and search a node.

1. Add () function add the node at the end of the linked list.
2. Delete () function delete the node from the end of the linked list.
3. Search() function, searches the specific key that is passed by the user at the runtime.

* For Vector:

I have made a template type vector so that we can deal with both characters and integers.

All the functions are written in vector like

1. vectorClass() is the Default constructor to initialise an initial capacity of 1 element and allocating storage using dynamic allocation
2. void push(T data) is Function to add an element at the last.
3. void pop() is the function to delete last element.
4. Search() function, searches the specific key that is passed by the user at the runtime.

Question 2:

I have made functions like:

* Recursive Binary Search:  
   A recursive binary search function. It returns location of x in given array arr[l..r] is present, otherwise -1.
* Iterative Binary Search:  
  A iterative binary search function. It returns location of x in given array arr[l..r] if present, otherwise -1.
* Linked-type binary search:  
  Function for implementing the Binary Search on linked list.

We can clearly see that iterative binary search run faster by testing it on arrays of sizes 1 million and 10 million with arrays that are filled with random numbers.

Question 3:

In this question, I have implemented bubble sort, insertion sort, and selection sort functions for 2d matrices.

I pass 2d matrix in the function of sort\_bubble(m), Then I sort the matrix row by row using bubble sort, insertion sort, and selection sort function in different calls.