**Problem Statement**

Write a prolog program to implement Quicksort which is a divide and conquer type algorithm for sorting.

**Design/Methodology/Algorithm/Data Structure**

Domains

list=integer

predicates

quicksort(list, list)

split(Integer, list, list, list).

concatenate(list, list, list).

printlist(list)

clauses

quicksort([],[])

quicksort([Head|Tail], sortedlist):-Split(Head,tail,Slist,Blist).

quicksort(Slist,Slist1)

quicksort(Blist,Blist1)

concatenate(Slist1, [Head|Blist], sortedlist)

printlist(sortedlist)

split(-,[],[],[])

Split(items,[Head1|Tail1],[Head1|Slist],Blist).

Item>Head1,1.

split(items,Tail, Slist, Blist)

split(items,Tail,Slist,Blist)

concatenate([], list, list)

concatenate([items|list1], list2, [items|list3]):-

concatenate(list1, list2, list3).

ptintlist([]):-n1

printlist([Head|Tail]):-write(Head,” “),printlist(Tail).

**Outputs/Results**

Quicksort([1,2,4,1,3,5,9,6],L)

1

3

6 9

3 6 9

3 4 5 6 9

1 2 3 4 5 6 9

L = [1,2,3,4,5,6,9]