**Day-13**

**Problem1**- [**21. Merge Two Sorted Lists**](https://leetcode.com/problems/merge-two-sorted-lists/)

Solution-

class Solution {

public:

ListNode\* mergeTwoLists(ListNode\* list1, ListNode\* list2) {

ListNode \*dummy = new ListNode(0);

ListNode \*temp = dummy;

while (list1 != nullptr && list2 != nullptr) {

if (list1->val < list2->val) {

temp->next = list1;

list1 = list1->next;

} else {

temp->next = list2;

list2 = list2->next;

}

temp = temp->next;

}

if (list1 != nullptr) {

temp->next = list1;

} else {

temp->next = list2;

}

return dummy->next;

}

};**Problem2**- [**23. Merge k Sorted Lists**](https://leetcode.com/problems/merge-k-sorted-lists/)

Solution-

class Solution {

public:

    ListNode\* mergeKLists(vector<ListNode\*>& list) {

      int n=list.size();

      priority\_queue<pair<int,ListNode\*>,vector<pair<int,ListNode\*>>,greater<pair<int,ListNode\*>>>pq;

      for(int i=0;i<n;i++)

      {

        if(list[i])

        {

            pq.push({list[i]->val,list[i]});

        }

      }

      ListNode \*dummy=new ListNode(-1);

      ListNode \*temp=dummy;

      while(!pq.empty())

      {

        auto it=pq.top();

        pq.pop();

        temp->next=it.second;

        if(it.second->next)

        {

            pq.push({it.second->next->val,it.second->next});

        }

        temp=temp->next;

      }

      return dummy->next;

    }

};

**Problem 3**- [**82. Remove Duplicates from Sorted List II**](https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/)

class Solution {

public:

ListNode\* deleteDuplicates(ListNode\* head) {

ListNode \*dummy=new ListNode(0,head);

ListNode \*prev=dummy;

while(head!=NULL)

{

if(head->next!=NULL && head->val==head->next->val)

{

while(head->next!=NULL && head->val==head->next->val)

{

head=head->next;

}

prev->next=head->next;

}

else

{

prev=head;

}

head=head->next;

}

return dummy->next;

}

};

**Problem 4**-  [[**143. Reorder List**](https://leetcode.com/problems/reorder-list/)[**21. Merge Two Sorted Lists**](https://leetcode.com/problems/merge-two-sorted-lists/)](https://leetcode.com/problems/palindrome-linked-list/)

class Solution {

public:

    void reorderList(ListNode\* head) {

        stack<ListNode\*>st;

        ListNode \*temp=head;

        while(temp!=NULL)

        {

            st.push(temp);

            temp=temp->next;

        }

        int size=st.size();

         temp=head;

         ListNode \*temp2=head->next;

         while(size >1)

         {

            temp->next=st.top();

            st.top()->next=temp2;

            st.pop();

            temp=temp2;

            temp2=temp->next;

            size=size-2;

         }

         temp->next=NULL;

         return;

    }

};