**Day-13**

**Problem1**- [**83. Remove Duplicates from Sorted List**](https://leetcode.com/problems/remove-duplicates-from-sorted-list/)

Solution-

class Solution {

public:

    ListNode\* deleteDuplicates(ListNode\* head) {

        if(head==NULL)return NULL;

        ListNode \* temp=head;

        while(temp!=NULL && temp->next!=NULL)

        {

            if(temp->val==temp->next->val)

            {

                temp->next=temp->next->next;

            }

            else

            temp=temp->next;

        }

        return head;

    }

};

**Problem2**- [**1290. Convert Binary Number in a Linked List to Integer**](https://leetcode.com/problems/convert-binary-number-in-a-linked-list-to-integer/)

Solution-

class Solution {

public:

int getDecimalValue(ListNode\* head) {

ListNode \*temp=head;

int n=0;

while(temp!=NULL)

{

n++;

temp=temp->next;

}

cout<<n;

temp=head;

int res=0;

while(temp)

{

// cout<<temp->val;

res+=(temp->val\*pow(2,n-1));

n--;

temp=temp->next;

}

return res;

}

};

**Problem 3**- [**203. Remove Linked List Elements**](https://leetcode.com/problems/remove-linked-list-elements/)

class Solution {

public:

    ListNode\* removeElements(ListNode\* head, int val) {

         if(head==NULL)return NULL;

          ListNode \*temp=head;

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

        {

          head=head->next;

        }

            while(temp!=NULL && temp->next!=NULL)

        {

            if(temp->next->val==val)

            {

                temp->next=temp->next->next;

            }

            else

            temp=temp->next;

            // cout<<temp->val;

        }

        return head;

    }

};

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

class Solution {

public:

bool isPalindrome(ListNode\* head) {

vector<int>v;

while(head!=NULL)

{

v.push\_back(head->val);

head=head->next;

}

int n=v.size();

cout<<n;

int flag=0;

for(int i=0;i<n/2;i++)

{

if(v[i]!=v[n-i-1])

return false;

}

return true;

}

};

**Problem 5**- [**70. Climbing Stairs**](https://leetcode.com/problems/climbing-stairs/)

class Solution {

public:

int solve(int n, vector<int>&dp)

{

if(n<=2)

return n;

if(dp[n]!=-1)return dp[n];

return dp[n]= solve(n-1,dp)+solve(n-2,dp);

}

int climbStairs(int n) {

// return climbStairs(n-1)+climbStairs(n-2);

vector<int>dp(n+1,-1);

return solve(n,dp);

}

};