	in the same of the
	22/04/2023
<u> </u>	Check your Linked List is palindrome or
ж.	
	1/p→ 1)→2→3→2→1→X
	0/b- True
	Palindrome -> Same when read either
	from left and right, then it is a palindrome.
	(2) (1)
	Approach-1
	Create a new linked list and that linked
L.	list will be the reverse of the input
	linked list and then compare both the
	linked list.
	Time complayible - O(p)
	Time complexity = O(n)  Space complexity = O(n)
	Space complexing on
	Approach-2
	Copy the data of linked list to the array
15 1	and then via 2 pointer approach, check
	for falindrome. However this again has
	yam has

## Scanned with Cam

PAY AND LONG	
A(Ay)	Space complexity of O(n). We need to do in
1247 2004 i	UCI) Space.
2 452	
	Approach-3
	r middle
	$(1) \rightarrow (2) \rightarrow (1) \rightarrow \times$
	Reverse the linked list from middle and for
* D	that we need to find the middle of the linked
1,27	list.
ર)	Mow start comparing both the half and if we
1	find any value not equal, simply return false.
1 1	Dothis until we reach NULL. Of false was not
	retwined then simply return true.
	middle marific Competent and
	$(1)\rightarrow(2)\rightarrow(3)$ / Move until
	temp
	templ
	Code
	AT I want become word with Hand march ()
	bool checkPalindrome (Node * & head) {
	// Empty linked list case
	if (head = = NULL) {
4.0	return true;
	2 dopore do
	//Single node is a falindrome
	if (head - next == NULL) ¿ To get n as middle
	xatuun trusi
C <sub>2</sub>	p fast = nead-next
	Done in previolass
	Node * middle = get Middle (head);  Node * middle = get Middle (head);  // Step-2 => Reverse the linked list after middle
	1/Stab-9 => Reverse the linked list after middle
计图185	11 5000

ocanneu with carn

## Done in LL class-2

	Node * reverse LL Head = reverse (middle+next)	
	middle - next = reverse LLHeadi // Optional	
	//Step-3 Compare both linked lists.	
	Node * temp = head;	
	Node * templ = head + next;	
	while (templ 1= NULL) {	
	Mot a palindrome	
1.32	if (temp -data 1 = temp1 -data) {	
	return false	
17. 7	1. 1. 3. 2. 1. 2. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.	
5%	temp = temp + next > 2 move forward	
_1,10, 2	templ = templ + next i Jin both linked	
-	3 . muri mudami la la manina lista manda de la	
	return true; // Palindrome	
	3	
	Time complexity = O(n)	
	Space complexity = 0(1)	
	9   1	
Q 2	Remove duplicates from sorted linked list.	
	$i/b \rightarrow 2 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 4$	
	0/p+ 1)-1 2-1 3-1 4)-1 X	
	THE TEXT	
	Approach-1	
	1 court	
The stand	$(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) $	
	July 1 of the second	
, - 1 , I	head he	
J. 415	compare curr + data and curr + next + data.	
	194 $110$ $194000$ $11700$ $110$ $110$ $110$ $110$	
	forward. If found equal, then move	
	D move	
	I 🗎	

		_
);	cwur + next to cwur + next → next.	= ـ
	1) - 2 × 2 -> 3 -1 4 -1 4 -> x	
	Delete this node	
		`
	Similarly we will be able to delete the node	
	with value/data - 4.	`
		\
	Codleggy volv og ag Lag al 201 2	`.
		`
	Void remove Duplicates (Node * & head) {	
	//Empty linked list if (head = = NULL) {	`
_	retwin i	
	3 //Single node can't be duplicate	
	if (head - next = = NULL) {	_ _;
	retwin	_ ;
	1/Call	;
	Node * cww = head;	:
	while (cur != NULL) {	
	While Lawre ! = 110000	
	if ((court next 1= NULL) && (court data	
	== curr + next + data)) {	'
	// Change the pointers	'
	Node & temb = CWUINEXED	
	$I = CARR \rightarrow CPX + DPX +$	<u> </u>
	1/ Delete node for not wasting memory	
>	// Delete node for not wasting memory temp-next = NULL;	
	delete tempi	
	3	

Scarnieu with Caril

	else {//Simply move curr forward
	cur = curr -next
	3
	3 staril april 7 13/ 1
	3
	7 7 1 2
	Time complexity = O(n) Space complexity = O(1)
	Space complexity = O(1)
Q3	Sort Os, Is and 2s in the linked list.
	1/b - 2 - 0 - 1 - 1 X
-	$0/p \rightarrow 0 \rightarrow 10 \rightarrow 10 \rightarrow 2 \rightarrow 2 \rightarrow x$
	· / Little of the second of th
	Approach - 1
1)	Simply store number of Os, Is and as
2)	in the linked list
~ <i>\</i>	Now first add Os, Is and 2s with the help of count stored.
	Time complexity - 0(n)
,	Time complexity = O(n) Space complexity = O(1)
عالج و	This approach has one issue that here data replacement is done but
	data replacement is done but in interview, The person might say that data
	The person might say that dat
	The person might say that data replacement
	O(1)
	Approach - 2.
	$(1)\rightarrow (2)\rightarrow (2)\rightarrow (0)\rightarrow (0)$
- N	1 temp
	Scarnieu With Co

Scarined with cam

1)	Create 3 dummy nodes
-	JzeroHead
	1 2 (-1) 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
	J One Head
	(-1) a contact the contact of
	I two Head
	(-1)
	temp = head;
	head = head + nexti
	temp - next = NULLi
	So after travoising fully, we get
	JZE10 Pail
_ ર)	$(-1) \longrightarrow (0) \longrightarrow (0) \longrightarrow \times \times$
	One Tail
	$(-1)\rightarrow (1)\rightarrow \times$
	twoTail
	$(-1) \rightarrow (2) \rightarrow (2) \rightarrow \times$
	Join all the linked list
	Remove all the dummy nodes
	1 head
	$\bigcirc \rightarrow \bigcirc \rightarrow$
	retwin the head.
	Code
	head) { head) {
	Node * sort Linked List (Node * & head) {
	// Creation of dummy nodes Node * zero Head = new Node (-1);
	Node * Zero Head = 116W Head
	Node * zero Tail = zero Head; Node * one Head = new Node (-1);
	Noae * one Head = 11Ev

Scarnieu with Caril

```
Node * one Tail = one Head;
Node * two Head = new Node (-1);
Node * two Tail = two Head;
//Traverse original linked list
Node * Cwur = head;
while (cover! = NULL) {
    // Handling node with O value
    if (curr - data = =0)
         // Isolate node
         Node * temp = curvi
        curr = curr - nexti
        temp = next = NULLi
       //Append O in the zero Head LL.
       zeroTall Inext = tempi
        zero Tail = tembi
  else if (cwr + data = = 1) {
      //Same steps for isolation of
      O node
     // Append I in the one Head LL
     one Tail - next = temp;
    one Tail = temp;
else {
      //Same steps for isolation of
     0 node
    // Append 2 in the two Head
    two Tail - next = temb;
    two Tail = temp
```

	// Join them and delete dummy nodes	
	// Modify one Head LL	
	Node * temp = one Head;	
	One Head = one Head + nexton	
	temb + next = NULLi	
	delete temps	
	// Modify two Head LL	
	temp = two Head;	
	twoHead = twoHead + next;	
1	temb-next = NULLi	
	delete temp;	
	//Join List	
	if (one Head ! = NULL) {	
	MoneHead LLis not empty	
	Zero Tail - next = one Head // Attach	
	if (two Head [ = NULL) { // two Head LL not	
	one Tail - next = two Head; empty	
	3 Connect	
	3	
	else { //one Head LL is empty.	
<i></i>	// Attach o LL and a LL	
	if (two Head [ = NULL) {	
<i></i>	zero Tail - next = two Head;	
<i></i>	3	
	$\frac{3}{3} + \frac{1}{3} + \frac{1}$	
	// Delete dummy node of O LL	
	temp = zeroHead;  TC = dinear	
	2010116456	
	temp + 11ext - 11000)	
	delete temp	
	// Return new head of LL	
	return zerottead	
- CAREA	3	
	" Scarineu Willi Carl	

Q4	Add 2 numbers represented by linked
	list
	1/þ→ 2→4)→×
	0/p→ 2→5→8→×
	Reverse both the linked list
<u> </u>	Add both the linked list
3)	Reverse the answer obtained & that
	is the answer.
_	Code
	Node * add (Node * 4hl, Node * 8h2) {
. ,	//Step 1 Reverse both LL
	h = reverse (h1)i
	h2 = reverse (h2);
	//Step 2 Add the LL
	Node * ans Head = NULL
	Node * ans Tail = NULL;
	int carry = 0;
	while (head! = NULL && head & 1 = NULL)
	// Fina Suin
	int sum = carry + h1 + data + h2 + data
	771 11100 33-7
	int digit = SUM 1/10)
9 =	// Find carry
<u> </u>	carry = sum/lo;
	Node * new Node = new N / ( )
	if (ans Head = = NULL) { Node (digit)
	//First node to insert
	Scarnieu Willi Ca
'	Scallieu Willi Ca

Scarineu with 🕭m

```
ans Head = new Node;
    ans Tail = new Node;
    else { // Not Ist node
       ans Tail - next = new Node;
       ans Tail = new Node;
hl= hl-next;
h2=h2+next>
// Ist LL is bigger
While (hl 1 = NULL) {
    int sum = carry + hl - data;
    int digit = sum 10 )
    carry = sum/10;
   Node * new Node = new Node (digit);
   ans Tail I next = new Node;
   ans Tail = new Node;
    hl=hl +next;
// Ind LL is bigger
While (hal = NULL) {
   1/Same steps followed above
// Handle the carry
while (carry ! = 0)
     int sum = carry;
    int digit = sum % 10 >
     carry = sum/10;
    Node * new Node = new Node (digit)i
    ons Tail → next = new Node;
    ans Tail = new Node;
```

ocanneu with can

	classmate
(96)	
1	

	No. of the second secon
	// Reverse the ans LL
	ans Head = reverse (ans Head)
	return ans Head;
	3
	about vo the compact the
Note-	reverse function has been discussed in
	class 2 of linked list.
	de pareto está
	Handle extra cases
	if (head1 = = NULL)
	retwin head 2;
	if (head2 = = NULL)
~-	return head 1
_	Call Value (Contraction)
	Time complexity = O(max(m,n)) - Linear Space complexity = O(1)
	Space complexity = O(1)
	Laboret Sur De Colon De Colon

Scarried with Cam