Node Node Node Node Ata: int next: Node* + gettern(: int + getData(int val): void + SetNext (Node*): Void + Node () Node **SetNext (Node*): Void + Node () **Node ** **ToddFront (int newThem): Void + Node () **SetNext (Node*): Void + Node (int newThem): Void + AddFront (int position, int newThem): void + search (int newThem): int + delete Front (): void + delete Front (): void + delete Front (): void + delete Front (int position): int + print Thems (): Void + print Thems (): Void + LinkedList (int array (): Int size)		Camlin Page Date 1 1
chata: int next: Node* + getData(): int + getData(): Noid + setNext (node* V): Void + setNext (Node* V): Void + Node() header: Node * + coddFroat (int newThem): Void + toddFroat (int newThem): Void + toddFroat (int position, int newTtem): Void + toddAtforition (int position, int newTtem): Void + search (int sexTtem): int + deleteFroat (): Void + deleteFroat (): Void + deleteFroat (int position): int + printTems(): Void + printTems(): Void + LinkedList (int owny [], int size)		UML DIAGRAM
chata: int next: Node* + getData(): int + getData(): Noid + setNext (node* V): Void + setNext (Node* V): Void + Node() header: Node * + coddFroat (int newThem): Void + toddFroat (int newThem): Void + toddFroat (int position, int newTtem): Void + toddAtforition (int position, int newTtem): Void + search (int sexTtem): int + deleteFroat (): Void + deleteFroat (): Void + deleteFroat (int position): int + printTems(): Void + printTems(): Void + LinkedList (int owny [], int size)		
10 Linkedlist tadderont (int resottern) & void + sear h (int resottern) & void + sear h (int resottern) & void + sear h (int position, int resottern) & void + delete Front () & void + delete Position (int position) & void + delete Front () & void + delete Front () & void + delete Position (int position) & void		Node
10 Linkedlist tadderont (int resottern) & void + sear h (int resottern) & void + sear h (int resottern) & void + sear h (int position, int resottern) & void + delete Front () & void + delete Position (int position) & void + delete Front () & void + delete Front () & void + delete Position (int position) & void		
10 Linkedlist tadderont (int resottern) & void + sear h (int resottern) & void + sear h (int resottern) & void + sear h (int position, int resottern) & void + delete Front () & void + delete Position (int position) & void + delete Front () & void + delete Front () & void + delete Position (int position) & void		datasint
+ Set Node * V) & void		
+ Set Node * V) & void	5	+getData() & int
+ Set Node * V) & void		+ Set Data (int val) & void
header & Node * header & Node * taddFront (int newTtern) & void taddFront (int newTtern) & void taddAtPosition (int position, int newTtern) & void t search (int resistern) & int t delete Front () & void t delete Front () & void t delete Position (int position) & void t get Item (int position) & int t print Items () & void t LinkedList (int army [], int size)		+ Set Next (Node* V) & Void
header & Node * header & Node * taddFront (int new Ttem) & void taddFront (int new Ttem) & void taddAt Position (int position, int new Ttem) & void t search (int new Ttem) & int t delete Front () & void t delete Front () & void t delete Position (int position) & void t get Ttem (int position) & int t print Ttems () & void t LinkedList (int army [], int size)		+ Node ()
header & Node * header & Node * taddFront (int newTtem) & vaid taddFront (int newTtem) & void taddAtPosition (int position, int newTtem) & vaid t sear h (int newTtem) & int t delete Front () & vaid t delete Front () & vaid t delete Position (int position) & void t get Ttem (int position) & int t print Ttems () & void t LinkedList (int army [], int size)		
header: Node * + add Front (int new Ttem): void + add End (int new Ttem): void + add Attosition (int position, int new Ttem): void + search (int mentem): int + delete Front (): void + delete Frad (): void + delete Position (int position): void + get Ttem (int position): int + print Ttems (): void + Linked List (int army [], int size)	10	
taddErant (int new Them) & void taddErad (int new Them) & void taddAtPosition (int position, int new Them) & vaid t search (int seas Them) & int t delete Front () & void t delete Frad () & void t delete Position (int position) & void t get Them (int position) & int t print Thems () & void t LinkedList (int army [] int size)		Linkedlist
taddFront (int new Them) & void taddEnd (int new Them) & void taddAtPosition (int position, int new Them) & void t sear h (int see Them) & int t delete Front () & void t delete Fnd () & void t delete Position (int position) & void t get Them (int position) & int t print Thems () & void t linked List (int army [] int size)		
taddEnd(int new) & void +addAtPosition(int position, int newTenn) & void + search(int sexTern) & int + deleteFront() & void + deleteFront() & void + deleteFosition(int position) & void + getTtern(int position) & int + printTterns() & void + LinkedList() + LinkedList(int army [], int size)		hender: Node *
taddEnd (int new) & void +addAtPosition (int position, int newTenn) & void + search (int sentem) & int + delete Front () & void + delete Fnd () & void + delete Position (int position) & void + getTern (int position) & int + print Thems () & void + LinkedList (int army [], int size)		taddFront (int new Them) & Void
+ addAt-Position (int position, int newTern) & void + search (int position) & int + delete Front () & void + delete Food () & void + delete Position (int position) & void + getTern (int position) & int + print Items () & void + LinkedList (int array [] int size)	15	taddEnd (int newsTem) & void
+ search (int see Item) & int + delete Frant () & void + delete Frant () & void + delete Position (int position) & void + get Item (int position) & int + print Items () & void + Linked List (int army [] int size) + Linked List (int army [] int size)	10	tadd At-Position (int position int new Them) & void
+ delete Frad () & vaid + delete Frad () & vaid + delete Position (int position) & vaid + get Them (int position) & int + print Thems () & vaid + Linked List (int army [] int size)		+ search (int protem) & int
+ delete Fnd () o void + delete Position (int position) o void + get Them (int position) o int + print Thems () o void + Linked List (int army [] int size)		
+ delete Position (int position) & void + get Them (int position) & int + print Thems () & void + Linked List (int army [] int size) + Linked List (int army [] int size)		= I
+ getTem (int position): int + print Thems () & void + LinkedList () + LinkedList (int army [], int size)		Late La Prailion (introposition) à Void
+ print Thems() & Void + LinkedList() + LinkedList(int army [] int size)	20	TOBLET SIMBLE TO SING
+ LinkedList (int army [] int size)		T get rem circ position
+ LinkedList (int army [] int size)		
		+ LinkedList()
		+ LinkedList Cint array 1 "11 Size)
·	25	
	Descript	Hon e-
Description o-		
Description :-	Nodo	
	17006	
Description:- Node	30)-> returns the value of the data member

4 2000

	Camlin Page Date				
*	getNeset() -> returns neset which is of type -				
*	SetData (int val) => The function sets the value of val into the data member data (i-e data = value).				
5	SetNesct (Node * V) -> The function sate Node * V into				
*	Mode () -) A default constructor of class Node that initializes the data variables of class Node.				
10	LinkedList				
	addfront (int newTtern) -> The function inserts a new				
	node, containing the newstrem, at the beginning of the list addlind (int newstrem) -> The function inserts a new node,				
15	add At-Pasition (int position, int newstern) is The function inserts a new node, containing the newstern, such that it is the				
	position-th member of the list. i.e we assume the 1st element of the list is in position 1. It position is larger				
20	than the size of the list, the new item is added to the end of the list. If position < 1, the new item is				
*	search (int item) -) The function searched the list for				
25	the item and found it both prints the position of the item (followed by a space) and returns the position of the item in the list. It not found both prints 0 and				
	deleteFront () -> The function deletes the 1st element of the list.				
	deleteted () -> The function deletes the last element of the list.				

delete Position (int position) > The function deletes the

element at the given position of the list. It the position all or larger than list size print "outside range".

· · · · · · · · · · · · · · · · · · ·		Camlin Page Date			
	getThem (int position) -> The	function both prints the value space and returns the value			
	of the item at the giver	position of the list. Tf			
5	limits Lint > := masco and	both prints std: numeric			
	I ''	Should add / include < limits>			
	print Tems () -) The tun	ction prints the value of the			
	items of the list from he nothing is printed.	and to tail - If empty list			
10	LinkedList () -> A constru	ictor with no parameters			
	which makes an empty lis	st.			
	LinkedList (intarroy of integers and makes a linked list				
	containing all the elements of the array in some				
15	order. The takes size	at array as 2nd parameter.			
	Test Cases				
20	Input	Output			
1)	52710 AP 39	529710			
2)	3421 DP 30	341			
3)	45 20 2 10 GT 3 0	2 45 20 2 10			
25					
30					