Introduction to linked lists Linked lists are similar to arrays (Linear data Structures) 7 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are Stored in Contiguous memory locations} 7 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are Stored in Contiguous memory locations} 17 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are Stored in Contiguous memory locations} 17 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are Stored in Contiguous memory locations} 18 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are stored in Contiguous memory locations. 19 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are not Stored in Contiguous memory locations. 10 11 12 18 22 \$\int_{\text{in}} Arrays clements are not Stored in Contiguous memory locations.	Similar to arrays (Linear data Structures) 22 Arrays clements are Stored in Configuous memory locations II	And the second
Linked lists are similar to arrays (Linear data Structures) 10 11 12 18 22 \$\infty \text{In Arrays clements are Stored in Configuous memory locations} \\ 10 11 11 NULL \$\infty\$ In linked lists; Clements are Stored in non Configuous memory locations Why Linked Lists? Memory and the capacity of an array remains fixed In case of linked lists, who can keep adding and removing elements without any capacity Constraints Drawbacks of linked lists Fixtra memory Space for pointers is required for every node 1 points is Random access not allowed as elements are not Stored in Contiguous memory locations. Implementation Linked list Can be implemented using a Structure in C language Struct Node \{ \text{int} data; \text{int} data;	Similar to arrays (Linear data Structures) 22 Arrays clements are Stored in Configuous memory locations II	
To 11 12 18 22 Arrays clements are Stored in Configuous memory locations	22 ⇒ In Arrays clements are Stored in Configuous memory locations → II → NULL ⇒ In linked lists, clement Lement are Stored in non Contiguo memory locations abacily of an array rumains fixed d lists, we can keep adding and without any capacity constraints lists for pointers is required for every node 1 points is need allowed as elements are not Stored in locations.	
To 11 12 18 22 Arrays clements are Stored in Configuous memory locations	22 ⇒ In Arrays clements are Stored in (ontiguous memory locations → II → NULL ⇒ In linked lists, clement lement are Stored in non Contiguo memory locations abacily of an array rumains fixed d lists, we can keep adding and without any capacity constraints lists for pointers is required for every node 1 pointes is need allowed as elements are not Stored in locations.	
Jaha Pointer to next element Are Stored in non Contents Why Linked Lists? Memory and the Cabacity of an array remains fixed: In Case of Linked Lists, we can keep adding and removing elements without any capacity Constraints Drawbacks of Linked Lists Extra memory Space for pointers is required (for every node 1 points is Random access not allowed as elements are not Stored in Contiguous memory locations. Implementation Linked List can be implemented using a Structure in C language Struct Node ?	Jement are Stored in non Contiguo memory locations abacity of an array remains fixed. A lists, we can keep adding and without any capacity Constraints lists for pointers is required for every node 1 pointes is need allowed as elements are not Stored in locations.	
Jaha Pointer to next element Are Stored in non Contents Why Linked Lists? Memory and the Cabacity of an array remains fixed: In Case of Linked Lists, we can keep adding and removing elements without any capacity Constraints Drawbacks of Linked Lists Extra memory Space for pointers is required (for every node 1 points is Random access not allowed as elements are not Stored in Contiguous memory locations. Implementation Linked List can be implemented using a Structure in C language Struct Node ?	Jement are Stored in non Contiguo memory locations abacity of an array remains fixed. A lists, we can keep adding and without any capacity Constraints lists for pointers is required for every node 1 pointes is need allowed as elements are not Stored in locations.	
Why linked lists? Memory and the capacity of an array rumains fixed. In case of linked lists, we can keep adding and removing elements without any capacity constraints. Drawbacks of linked lists. Textra memory space for pointers is required for every node 1 points is Random access not allowed as elements are not stored in contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language Structure in the data;	abacity of an array rumains fixed. A lists, we can keep adding and without any capacity Constraints Lists for pointers is required for every node 1 pointers is need allowed as elements are not Stored in locations.	
Why linked lists? Memory and the capacity of an array remains fixed. In case of linked lists, we can keep adding and removing elements without any capacity Constraints. Drawbacks of linked lists. Extra memory Space for pointers is required for every node 1 points is a Random access not allowed as elements are not Stored in Contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language. Shuct Node? int data;	abacity of an array rumains fixed. A lists, we can keep adding and without any capacity Constraints Lists for pointers is required for every node 1 pointers is need allowed as elements are not Stored in locations.	
Why Linked Lists? Memory and the capacity of an array rumains fixed. In case of linked lists, we can keep adding and removing elements without any capacity constraints Drawbacks of linked lists Extra memory space for pointers is required (for every node 1 points is Random access not allowed as elements are not stored in contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language struct Node? int data;	abacity of an array rumains fixed. I lists, we can keep adding and without any capacity constraints lists for pointers is required for every node 1 pointes is need allowed as elements are not stored in locations.	
Memory and the capacity of an array rumains fixed. In case of linked lists, we can keep adding and removing elements without any capacity constraints Drawbacks of linked lists Textra memory Space for pointers is required for every node 1 points is Random access not allowed as elements are not stored in contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language structure in the contiguous interest and the contiguous continuation.	abacity of an array rumains fixed. I lists we can keep adding and without any capacity constraints lists for pointers is required for every node 1 points is near allowed as elements are not stored in locations.	
Memory and the capacity of an array rumains fixed. In case of linked lists, we can keep adding and removing elements without any capacity constraints Drawbacks of linked lists Textra memory Space for pointers is required for every node 1 points is Random access not allowed as elements are not stored in contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language structure in the contiguous interest and the contiguous continuation.	abacity of an array rumains fixed. I lists we can keep adding and without any capacity constraints lists for pointers is required for every node 1 points is near allowed as elements are not stored in locations.	15
Drawbacks of linked lists Textra memory Space for pointers is required for every node 1 points is Random access not allowed as elements are not Stored in Contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language Struct Node ? int data;	for pointers is required (for every node 1 pointes is need allowed as elements are not Stored in locations.	
Drawbacks of linked lists Textra memory Space for pointers is required for every node 1 points is Random access not allowed as elements are not Stored in Contiguous memory locations. Implementation Linked list can be implemented using a Structure in C language Struct Node ? int data;	for pointers is required (for every node 1 pointes is need allowed as elements are not Stored in locations.	N. A.
Implementation Linked list can be implemented using a Structure in Clanguage Struct Node { int data;	locations.	
Implementation Linked list can be implemented using a Structure in Clanguage Struct Node { int data;	locations.	
Implementation Linked list can be implemented using a Structure in Clanguage Struct Node { int data;	locations.	\rightarrow
Implementation Linked list can be implemented using a Structure in Clanguage Struct Node { int data;		<u></u>
Linked list can be implemented using a Structure in C. language Struct Node { int data;	implemented using a Structure in Clanguage	
Linked list can be implemented using a Structure in C. language Struct Node { int data;	implemented using a Structure in C language	
int data;		
int data;		
Struct Node * next; => Self refrencing Structure	next; => Self refrencing Structure	
3;	' ' '	