## Classmate Date Page 51

## Linked List

DP.	Pir	1ti	00
The.	11	111	7.1

- -> An order collection of homogenous deta
- -> where elements can be added anywhere and removed from anywhere

Catagory	Array	Linked List
	Contiguous black od	Nodes with
	memory !	Pointers
Dynamiz	Fixed size unless	Easily grows
size	manually resized	orshinks
	Ine Sticient dor	
	Arequent insertions	
	/removals.	-Ovals
Memory	may lead to	Reduces
 Allocation	fragmentation	frugmentation
Use	Fast access with	No fixed size
Cuse	Known Size	Fapquent in sostien frema

## Types

## ) Single linked List

- -> Can be traversed on y in one direction
- > Sorted LL
- . Flement are inserted in sorted order
- · peletion as per requirements.
- -> Unsorted by
- · Elements are inserted & deleted as per user Veguirements.

JImplementation

SNT

NAIgo LL Type Create Linkeu List ()

head \$= NULL;

2.11 2 anii Lorra Lines

2) LI Type Insert (LL Type Mead, Node Type New Mode)
-> & 1/Unsorted link key)

id (head == NULC) 12 bax in simply of

New Node > Next = NULL;

CHORS REALINGERHORS IN SPIE STORE

Case 1: insertion before head New Node > next = heads

head = New Node

Case 2: insertion after the last

Struct Node Type \*temp;

temp= head

While (temp > next 1= NULL)

temp=temp=next;

4cmp -> next = New Node;

case 3: insertion before some node ( Rey)

Struct NodeType\*tempy ...

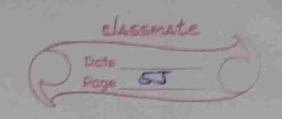
temps head; prevenully

while (temp -> data 1 = Kes as temp 1 = NULL)

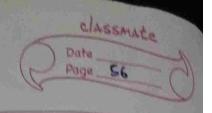
prev = temp ) temp = temp > next;

```
"d (temp == NULL) " Key not Found"
EISE
  New Node -> next = temp;
  prev I next : temp;
Casey: Presention after some node (key)
Struct Node Type #temp
temps Head; Prevanut,
while (temp =) data 1= key & & (empto)
      temp (=NULL)
temps temp onexts
12 (temp == NULL) " Rey not Found"
else
New Node -> next = temp > next;
temponext = New Node;
11 sorted
id ( LL = NULL) // directement
LL : Now Node; exit(0);
tempell, prove Null;
While ((NewNode -> data >temp> >data) &&
     temp ( = NULL)
Sprev = temps
temps temps noxy;
Pd (demp == NULL) 11 end
Prev -> next = New Node;
id (temp==LL && Newrode => date & temp => date)
```

	E NewNode > next = temp;
	LL=NewNode;
	3 - 1000 1 - 1090 a subola 64914 -
	else and transferrance
(227/22)	New Node -> next = temps
	prev -> next = New Node;
	3 12 4 4 4 6 (-1) 6 3 5 4 9 6 9 4
3)	Algo ET Delete (LLType Head, ET ele)
	gd int key)
	id (Head == NULL) = 9 )
4	Print Vunderflow 113 = / 131) 69
	exit; ( what to
	else and exact a bounder
	2 - TOWNSON - HAS OF THE MIST
1)	Search for element
	does not exist / 50+10211 @
3)	in Unsested List
47	in sorted list of ( and )
	3 (6) 1 is stouright 1
	3 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
Mark of	Deletions of shall a sportous (1) grinch
	1 Cojun : 19m3/
	id (Mead == NULL)
	2 Point 4 Under Slow 1, exit(); 3
	else ? 11 senrich
	Temp = Mead , Prev = NULL;
	while (Temp > data a key)
(#G), 0-1	prevo Temp; Temp= Temp. > Next;



i) (Temp > data 1 = key) 11 not found Error", Exito); & Elspid (temp -> data = = Key && prev == NULY) Head = Head Inext; 11 first elemens else id (temp) data == key letemp-)nex == NULL) // only element in LE HEARD = NULL else prev-) next = Temp-) next return Temp;



2)	Circular Linked List
-5	Last node is connected to first node
-7	It supports traversing from the end after
100	list to the beginning by making to
	node point back to the head of the !!
>	17 Implementation
	Reser (1)
2)	LL Type Insert [CLI Type head, Wode Type Name
	id (maid == NULL) // top
	Head = New Node !
	read > next = Head;
	else
	New Node -> Next = Head -> next,
	Head > next = value.
Tari	3
	1 middle
	1
	Create Node (temp) = Head)
	some cremp > element 1 = whicker se country
	2 1= Head)
	id (corrent see
	Newwode = Key) &
	Newwode Snept - Corrent Snept  Current Snept - New Node;
	3 - New Node,

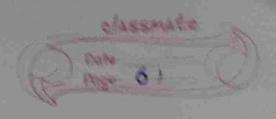
llend create Node (temp) = Head; while (temp > next 1= head) ? Current = current > next; 3 (Wrent - next = New Node; NewNode > Nex+ = head; 3 Algo ET Delete (CLL Type Head, ET ele) Hint Key ) & 11top id (head == NULL) "Underflow ! id ( head -> next == head) temp= tread; Mead = NULLS else Temp = head -> next; nead > next = temp -> next. return (tempodata) HEAd ER HEND Temp = head, Prev=NULL; while (temp -> next != hand) ? Prev = Temps Temp = Temp > next;

& Umiddle Temp = heads Prev= NULLs While (Temp->ele ) = key && temp-) newson ¿ prev= temp; temp= temp= next, data id (Temp + element = = key) & Prev-) next = current-) next; return (temps duta) 9 90 TI 0014 Under flowed I hend - ment == hend) +Cmp- "Deald: resurn term andeles 36 HEND 1810 12 1054 1304 NOVEL while (brain and to hear) ? Peter Mane: Temperatures

3	Doubly Linked List
-9	Can be traversed in both question direction
-7	Implementation
	Struct Node?
	intologias
	Node * next;
	Node* prev;
	3; The transcence bear a to to a good to
	Node + hend, List1, List2;
1)	Search
	temp= Head;
	While (New Node -) data 7 temp -> data &&
	temp-> next = NULL)
	temp=temp=next;
2)	WInsertion 1994
1.00	Create Node (New Node)
A)	First Node
	if (Head == NULL)
	Head = Neu Noue;
3)	At end
	if ( New Node > data > temp > dada & & temp > next
	== NULL)
4. 165	{ New Node > prev + temp;
	temp = next = NewNode;
	3
c)	Before head Node
	id (New Node -> claya (+emp -> data dd +emp == head)
	¿ New Node & next = temp;

100

temp-sprev = NewNode; 11 11 dung head = New Node; D) in between NewNode >next = temp; New Node > Prev = temp > Prev; temp > prev -> next = NewNode, temp-) preveneunodes 3) Deletion 1) if (Head 2= NOLL) 6 - 304 (394) 9/14() "Underflow 16 4 - 1 toon agreed by 1 search 1297 can mat a graph temp = head while (temp) = NULL 28 temp > data ( key) temp=+emp=>nex+; 3604 62x61 (A 11 uns. u ruecesdul search id (temp-7data > key 11(temp) next == NULL && temp > data < key) Error W. Compt. CAtaba Showard !! 2) Only Node. id (temp >data == key & & temp == head & & temp - JUGAT == MORTED = HABUE JUST of Head = NULL; return (temp) 3 south book souther ( Pd (temp>data == Key Al-temp> nox+ == NULL)



& temps preces next encer

\*\*Ctorn(temp)

4

4) General

2 temp -> next -> prev -- temp -> pext;

4 temp -> prev -> pext = + emp -> pext;

5) First node

id (femp == head && temp-soluta == rev)

& head= head > next;

femp = next > Brev = NULL;

vetorn (temp)

	Advantage
Total	Overflow can never occur unless omemory act
4	Juli.
->	Insertions & deletions are easier + hanto
	Contigoous (array) lists.
->	with large records, moving pointers is easier
	and faster than moving the items themselves
	Shon taille
	Disadvadages 188 book = 9 mo +)
-)	Pointers require extra space
-	Linked Lists do notallow random access
	Time must be spent traversing & changing
->	Program 1
	Programming is trickier & with pointer