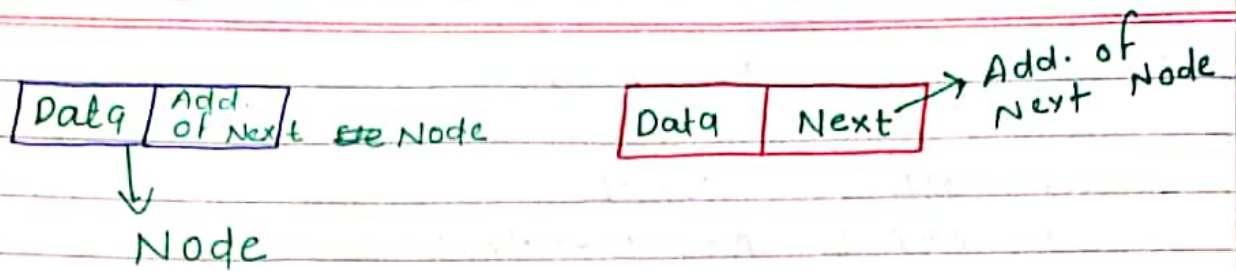
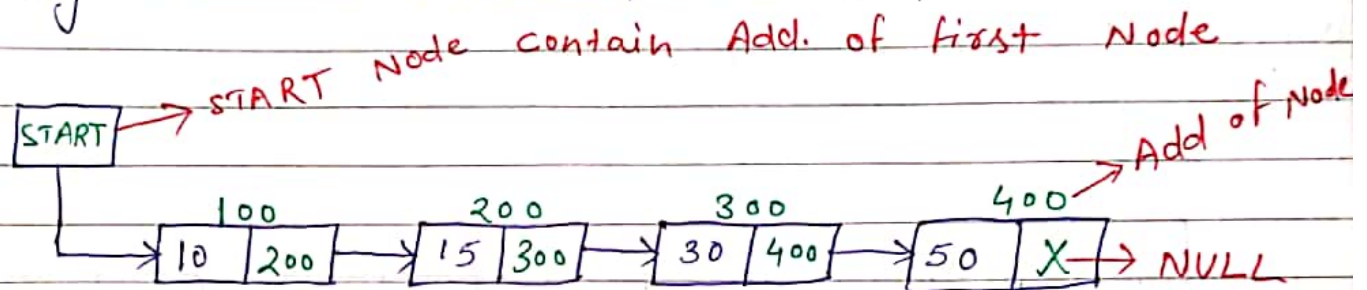


Linked List



Singly Linked List

1. Algo to traverse a Linked List.



- (i) Set PTR = START
- (ii) Repeat Step (iii) and (iv), while PTR != NULL
- (iii) Apply process on PTR -> data.
- (iv) Set PTR = PTR -> Next
- (v) Exit

2. Algo to insert an Element in the beginning

AVAIL -> AVAIL is a pointer which contains / points the free blocks of memory.

OR

AVAIL is the Linked List of free memory blocks.

checking Available Space in memory.

(i) If $AVAIL = NULL$, Write overflow & Exit

(ii) Set $NEW = AVAIL$

Set $AVAIL = AVAIL \rightarrow \text{Next}$ // Switching $AVAIL$ to
Next free memory
Block for later use.

(iii) Set $NEW \rightarrow \text{Next} = \text{START}$

(iv) Set $\text{START} = NEW$

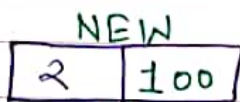
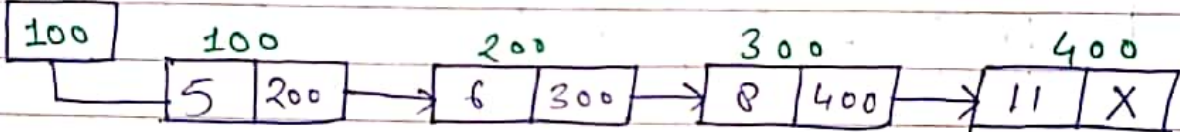
(v) Set $NEW \rightarrow \text{data} = \text{ITEM}$

(vi) Exit

Item is element which is inserted
in new

e.g.

START

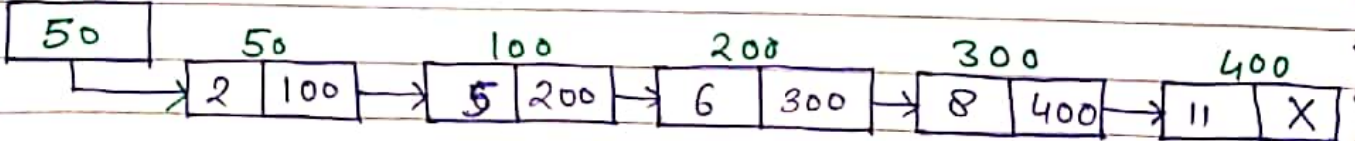


50 → Address of NEW Node

$NEW \rightarrow \text{data} = 2$

$NEW \rightarrow \text{Next} = \text{START}$

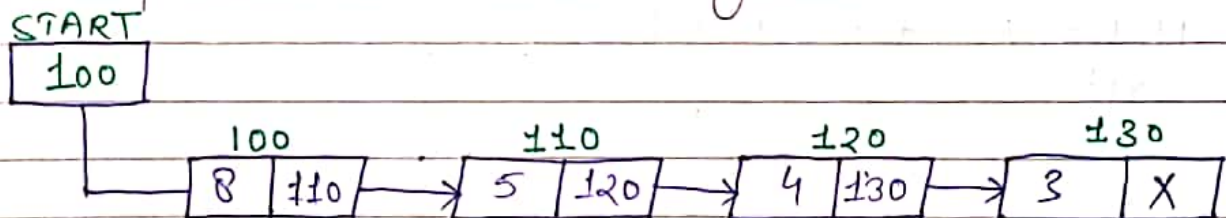
START



3. Algo to Insert an Element after the particular location by Value.

- (i) If $AVAIL == NULL$, write overflow & Exit.
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$
- (v) Set $PTR = START$
- (vi) Repeat Step (vii) While $PTR \rightarrow data \neq VALUE$
- (vii) Set $PTR = PTR \rightarrow Next$
- (viii) Set $NEW \rightarrow Next = PTR \rightarrow Next$
- (ix) Set $PTR \rightarrow Next = NEW$
- (x) Exit

4. An Algo to insert an Element before a particular location by Value.



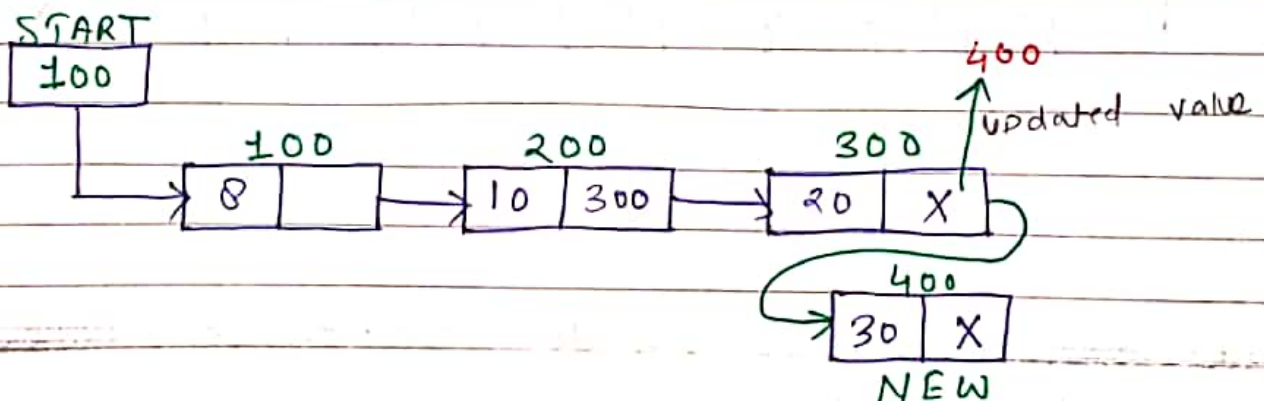
→ Inserting Element before third Node which have value/data 4

- (i) If $AVAIL == NULL$, write overflow & Exit.
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$
- (v) Set $PTR = START$
- (vi) Set $PPTR = PTR$

- (vii) Repeat Step (viii) and (ix) while $PTR \rightarrow data \neq 4$
- (viii) Set $pptr = PTR$
- (ix) Set $PTR = PTR \rightarrow Next$
- (x) Set $pptr \rightarrow Next = NEW$
- (xi) Set $NEW \rightarrow Next = pptr$
- (xii) Exit

5. Algo to Insert Node At the End.

- (i) If $AVAIL == NULL$, write overflow & exit.
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $PTR = START$
- (v) Repeat Step (vi) while $PTR \neq NULL$
- (vi) $PTR = PTR \rightarrow Next$
- (vii) $PTR \rightarrow Next = NEW$
- (viii) $NEW \rightarrow Next = NULL$
- (ix) Exit



6. Algo to delete first Node of Linked List.

- (i) If $START == NULL$, Write underflow & Exit
- (ii) Set $PTR = START$
- (iii) Set $START = PTR \rightarrow Next$
- (iv) ~~Exit~~ Free (PTR)
- (v) Exit

7. Algo to delete last Node of Linked List.

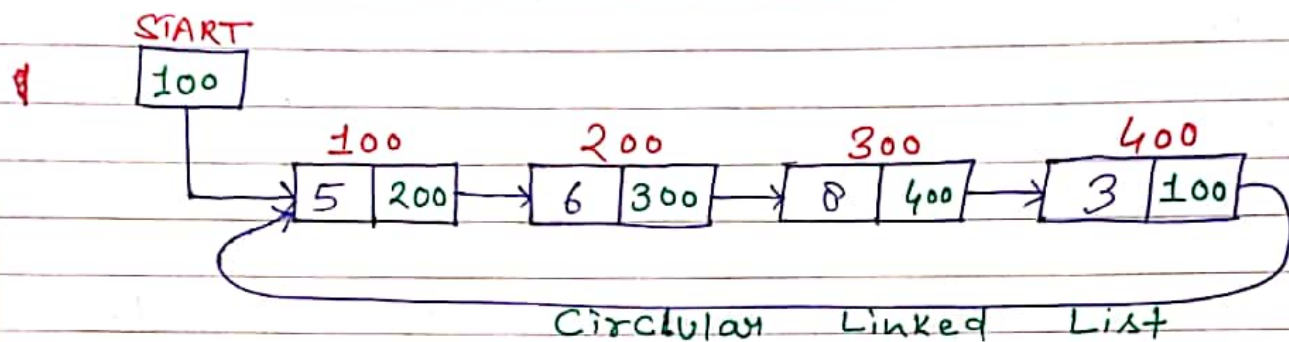
- (i) If $START == NULL$, Write underflow & Exit.
- (ii) Set $PTR = START$
- (iii) Set $PPTR = START$
- (iv) Repeat Step (v) & (vi) While $PTR \rightarrow Next \neq NULL$.
- (v) Set $PPTR = PTR$
- (vi) Set $PTR = PTR \rightarrow Next$
- (vii) Set $PPTR \rightarrow Next = NULL$
- (viii) Free (PTR)
- (ix) Exit

8. Algo to delete a Node from Linked List by value / data.

- (i) If $START == NULL$, write underflow & Exit.
- (ii) Set $PPTR = PTR = START$
- (iii) Repeat Step (iv) & (v) While $PTR \rightarrow data \neq VALUE$
- (iv) Set $PPTR = PTR$
- (v) Set $PTR = PTR \rightarrow Next$

- (vi) Set $PTR \rightarrow Next = PTR$
- (vii) Exit

Circular Linked List



1. Algo to traverse Linked List

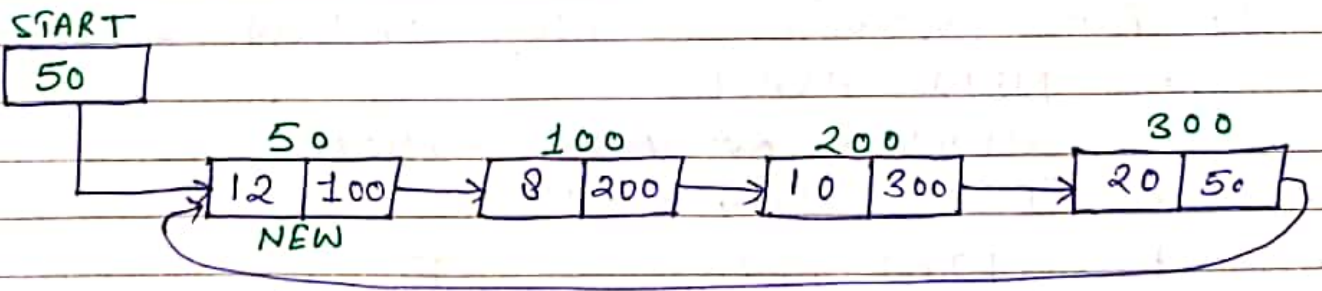
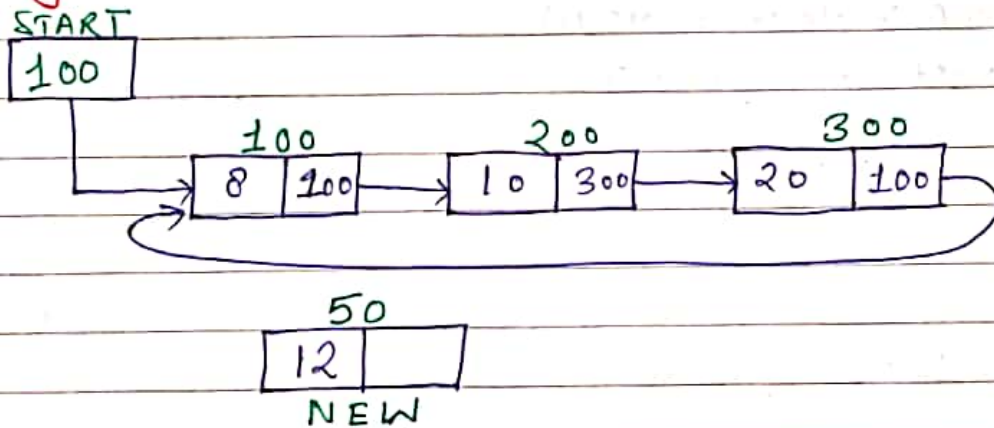
- (i) IF $START == NULL$, Write Underflow & Exit
- (ii) Set $PTR = START$
- (iii) Repeat Step (iv) & (v) while $PTR \rightarrow Next \neq START$
- (iv) Apply process on $PTR \rightarrow data$
- (v) Set $PTR = PTR \rightarrow Next$.
- (vi) Exit.

2. Algo to insert an Element/Node at the Beginning.

- (i) IF $AVAIL == NULL$, write overflow & Exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$

- (iv) Set $NEW \rightarrow data = ITEM$
- (v) Set $PTR = START$
- (vi) Repeat Step (vii) while $PTR \rightarrow Next \neq START$
- (vii) Set $PTR = PTR \rightarrow Next$
- (viii) Set $PTR \rightarrow Next = NEW$
- (ix) Set $NEW \rightarrow Next = START$
- (x) Set $START = NEW$
- (xi) Exit.

E.g.



3. Algo to Insert Element at the End of Linked List.

- (i) IF $AVAIL == NULL$, write overflow & Exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$.
- (v) Set $PTR = START$
- (vi) Repeat Step (vii) While $PTR \rightarrow Next \neq START$
- (vii) $PTR = PTR \rightarrow Next$
- (viii) Set $PTR \rightarrow Next = NEW$
- (ix) $NEW \rightarrow Next = START$
- (x) Exit.

4. Algo to Insert an Element/Node before a particular location by data/value.

- (i) IF $AVAIL == NULL$, write overflow & Exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = ~~AVAIL~~ AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$
- (v) Set $PTR = pPTR = START$
- (vi) Repeat Step (vii) & (viii) While $PTR \rightarrow data \neq VAL$
- (vii) Set $pPTR = PTR$
- (viii) Set $PTR = PTR \rightarrow Next$
- (ix) Set $pPTR \rightarrow Next = NEW$
- (x) Set $NEW \rightarrow Next = PTR$
- (xi) Exit

5. Algo to Insert a Node After the particular location by value/data.

- (i) If $AVAIL == NULL$, Write overflow & Exit.
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$.
- (v) Set $PTR = START$
- (vi) Repeat Step (vi) while ~~$PTR \rightarrow Next$~~ $PTR \rightarrow data \neq VAL$
- (vii) Set $PTR = PTR \rightarrow Next$
- (viii) Set $NEW \rightarrow Next = PTR \rightarrow Next$
- (ix) Set $PTR \rightarrow Next = NEW$
- (x) Exit

6. Algo to delete first Node of Linked List.

- (i) If $START == NULL$, Write overflow & Exit
- (ii) Set $ppTR = PTR = START$
- (iii) Repeat Step (iv) while $PTR \rightarrow Next \neq START$
- (iv) Set $PTR = PTR \rightarrow Next$
- (v) Set $PTR \rightarrow Next = START \rightarrow Next$
- (vi) Set $START = START \rightarrow Next$
- (vii) Free (ppTR)
- (viii) Exit

7. Algo to delete Last Node of Circular Linked List.

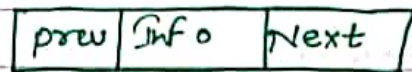
- i) If $START == NULL$, Write overflow & Exit.
- (ii) Set $PPTR:PTR = START$
- (iii) Repeat Step (iv)&(v) While $PTR \rightarrow Next \neq START$
- (iv) Set $PPTR = PTR$
- (v) Set $PTR = PTR \rightarrow Next$
- (vi) Set $PPTR \rightarrow Next = START$
- (vii) $Free(PTR)$
- (viii) Exit

8. Algo to Delete a Node from Circular Linked List by data/value.

Same As Singulay Linked List.

Double Linked List or

Two way Linked List

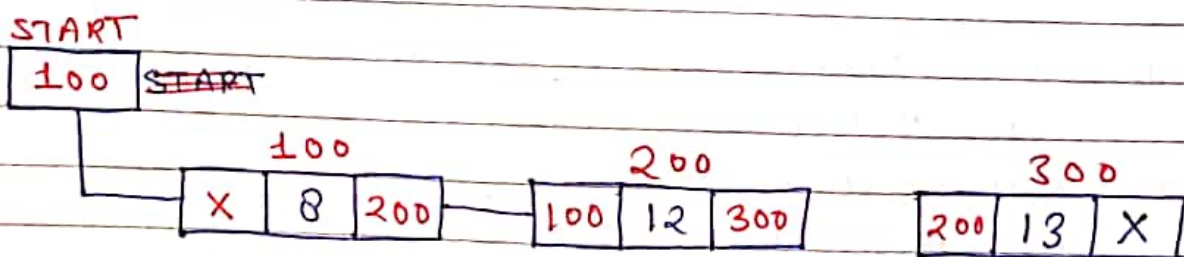


→ The Information field Info stores the data of a Node.

→ The pointer field Next contains the Address of Next Node

→ The pointer field prev contains the Address of previous Node.

1. Algo To traverse double Linked List.



(i) Set PTR = START

(ii) Repeat step (iii) and (iv) while PTR != NULL

(iii) Apply process on PTR → data

(iv) Set PTR = PTR → Next

(v) Exit

2. Algo to Insert an Element in the beginning.

- (i) If $AVAIL == NULL$, write overflow & exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow \text{Next}$
- (iv) Set $NEW \rightarrow \text{data} = \text{ITEM}$
- (v) Set $NEW \rightarrow \text{Next} = \text{START}$
- (vi) Set $NEW \rightarrow \text{prev} = NULL$
- (vii) Set $\text{START} \rightarrow \text{prev} = NEW$
- (viii) Set $\text{START} = NEW$
- (ix) Exit.

3. Algo to Insert an Element At the End.

- (i) If $AVAIL == NULL$, write overflow & exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow \text{Next}$
- (iv) Set $PTR = \text{START}$
- (v) Set $NEW \rightarrow \text{data} = \text{ITEM}$
- (vi) Repeat Step (vii) while $PTR \rightarrow \text{Next} \neq NULL$
- (vii) Set $PTR = PTR \rightarrow \text{Next}$
- (viii) Set $PTR \rightarrow \text{Next} = NEW$
- (ix) Set $NEW \rightarrow \text{prev} = PTR$
- (x) Set $NEW \rightarrow \text{Next} = NULL$
- (xi) Exit

4. Algo to Insert An Element After a particular location by data.

- (i) If $AVAIL == NULL$, write overflow & exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$, set $PTR = START$
- (v) Repeat Step (vi) While $PTR \rightarrow data \neq DATA$ (of Node)
- (vi) Set $PTR = PTR \rightarrow Next$
- (vii) $NEW \rightarrow Next = PTR \rightarrow Next$
- (viii) $NEW \rightarrow prev = PTR$
- (ix) $PTR \rightarrow Next \rightarrow prev = NEW$
- (x) $PTR \rightarrow Next = NEW$
- (xi) Exit

5. Algo to Insert An Element Before a particular location by Data.

- (i) If $AVAIL == NULL$, write overflow & exit
- (ii) Set $NEW = AVAIL$
- (iii) Set $AVAIL = AVAIL \rightarrow Next$
- (iv) Set $NEW \rightarrow data = ITEM$
- (v) Re. Set $PTR = START$
- (vi) Repeat Step (vii) While $PTR \rightarrow data \neq Data$ (of Node)
- (vii) $PTR = PTR \rightarrow Next$
- (viii) Set $PTR \rightarrow prev \rightarrow Next = NEW$
- (ix) Set $NEW \rightarrow prev = PTR \rightarrow prev$
- (x) Set $PTR \rightarrow prev = NEW$
- (xi) Set $NEW \rightarrow Next = PTR$
- (xii) Exit

6. Algo to delete first Node / Element

- (i) If $START == NULL$, write Underflow & Exit
- (ii) Set $PTR = START$
- (iii) Set $START = START \rightarrow next$
- (iv) Set $START \rightarrow prev = NULL$
- (v) Free (PTR)
- (vi) Exit

7. Algo to delete Last Node of Linked List.

- (i) If $START == NULL$, write Underflow & Exit
- (ii) Set $PTR = START$
- (iii) Repeat Step (iv) while $PTR \rightarrow next \neq NULL$
- (iv) Set $PTR = PTR \rightarrow next$
- (v) Set $PTR \rightarrow prev \rightarrow next = NULL$
- (vi) Free (PTR)
- (vii) Exit

8. Algo to delete Element by data.

- (i) If $START == NULL$, write Underflow & Exit
- (ii) Set $PTR = START$
- (iii) Repeat Step (iv) while $PTR \rightarrow data \neq DATA$
- (iv) Set $PTR = PTR \rightarrow next$
- (v) Set $PTR \rightarrow prev \rightarrow next = PTR \rightarrow next$
- (vi) Set $PTR \rightarrow next \rightarrow prev = PTR \rightarrow prev$
- (vii) Free (PTR)
- (viii) Exit