Doubly Link List

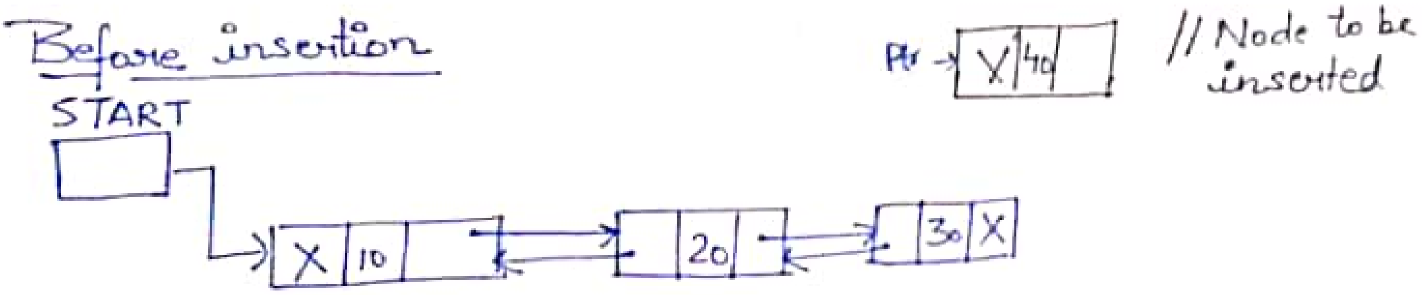
```
#include<stdio.h>
#include< malloc.h>
typedef struct nodetype          // Declaration of structure
{
   struct nodetype *prev;
    int info;
   struct nodetype *next;
  }node;
 node *start=NULL;
 Void create( )
 {
   node *temp, *ptr;
   int ch;
 temp=(node *)malloc(sizeof(node));
 Print("\n Input first node information");
  Scanf("%d", &temp->info);
  temp->prev=NULL;
   Start=temp;
  do
   {
   ptr=(node *)malloc(sizeof(node));
  printf("\n Enter info of next node");
   Scanf( "%d",&ptr->info);
  temp->next=ptr;
  ptr->prev=temp;
   temp=ptr;
 Printf("\n Enter choice except 1\t");
   Scanf("%d",&ch);
 }while(ch!=1);
 temp->next=NULL;
}

 void ftraverse()
  {
    node *ptr;
   printf("\n Forward traversing\t");
   ptr=start;
   while(ptr!=NULL)
    {
     Printf("%p|",ptr->prev);
     printf("%d |", ptr->info);
     Printf("%p--->", ptr->next);
     ptr=ptr->next;
    }
  }
   main()
   {
     create();
     ftraverse();
   }
```

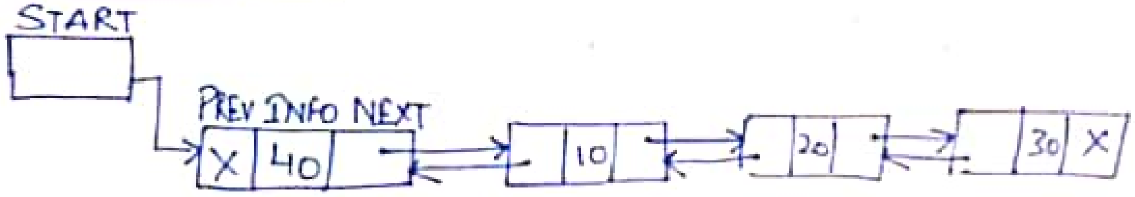
Doubly Linked List:-

Algorithm to insert an element in the beginning of the list:-

Before insertion



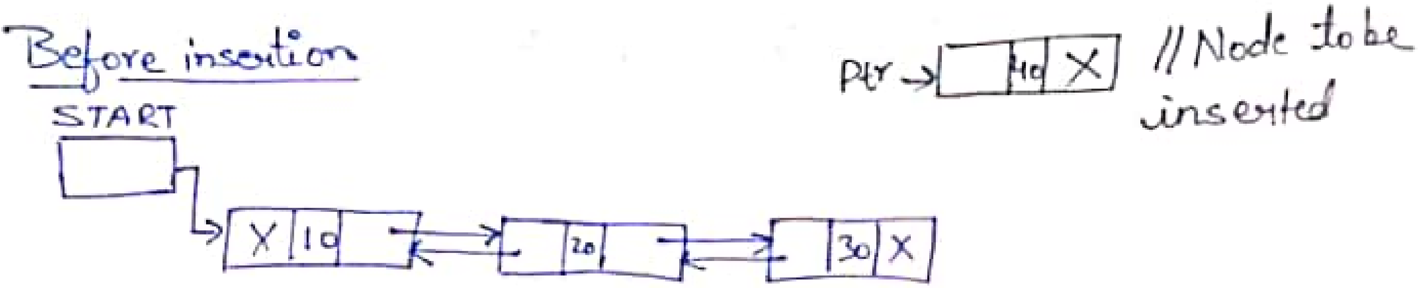
After insertion



```
Insertatbeg(INFO, PREV, NEXT, START, ELEMENT)
1) Create a node and address is assigned to Ptr.
2)    if( Ptr == NULL)
      Write: Overflow and Exit
3) Set INFO[Ptr] = ELEMENT
4)  if(START == NULL)        //No node from before in list
      Set PREV[Ptr] = NULL   //then the new node created
      Set NEXT[Ptr] = NULL     becomes first as well as
        Set START = Ptr        last node.
   else
   Set PREV[START] = Ptr     // if above two condition
   Set NEXT[PLr] = START        fail, and there are nodes
                                in list
      Set START = Ptr
      Set PREV[START] = NULL
5) Exit.
```

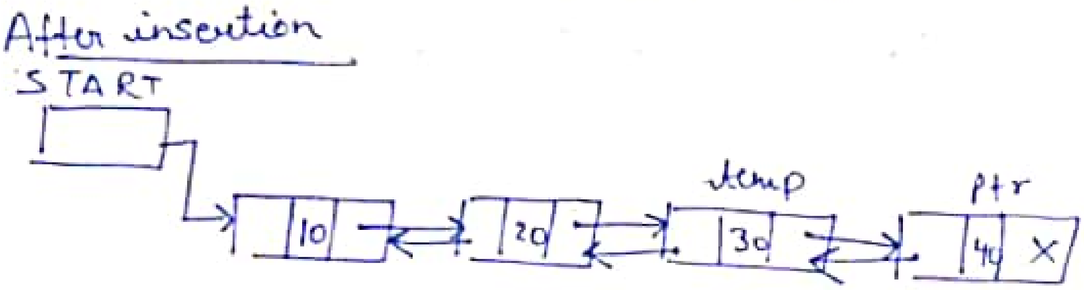
Algorithm to insert an element in the end of the list:-

Before insertion



// Node to be inserted

After insertion

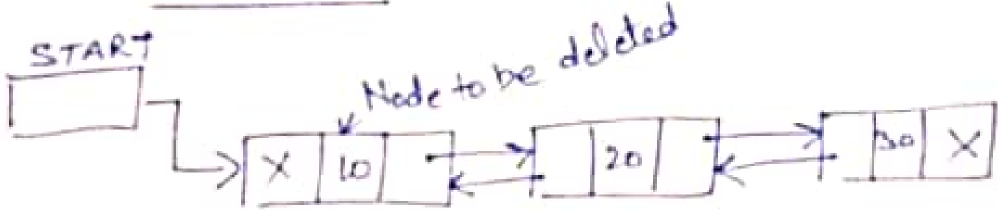


```
InsertatendDL( INFO, PREV, NEXT, ELEMENT)
1) Create a node and address is assigned to Ptr.
2) if ( Ptr== NULL)
       Write: Overflow and Exit
3) Set INFO [Ptr]= ELEMENT
4) if (STARТ== NULL)   // No node before in list
       Set PREV [Ptr]= NULL
       Set NEXT [Ptr]= NULL
       Set START = Ptr
   else
```

```
Set temp = START        // to traverse till end
While (NEXT[Temp] != NULL)
 {
  Set Temp = NEXT[Temp]
 }
 Set PREV[Ptr] = Temp
 Set NEXT[Temp] = Ptr
 Set NEXT[Ptr] = NULL
5) Exit .
```

Algorithm to delete a node in the beginning of the list :-

deletion

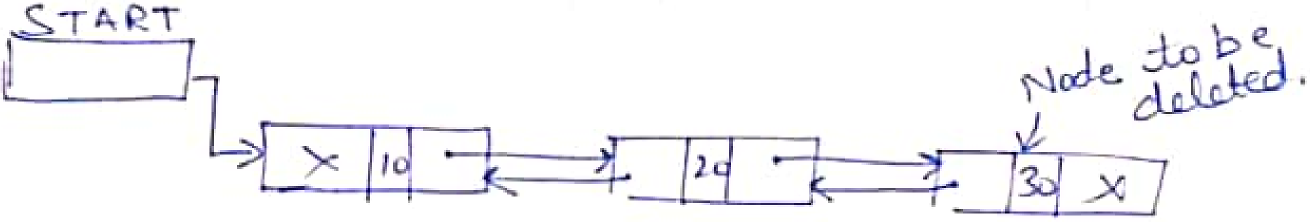


```
Deleteatbeg(INFO, NEXT, PREV, START)
1) if (START == NULL)
     Write: ~~ER~~ Underflow and
            Exit
2) Set temp = START
3) if (NEXT[START] = NULL)   // List contain a
                             Single node.
       Set START = NULL
   else
```

```
    Set START = NEXT[START]
     Set PREV[START] = NULL      // Now the next
                                   node in list's previous
                                 pointer field will become
                                  NULL, Since it is the
                                  first node of Doubly
                                  Link list.

4) Set K = INFO[Temp]
5) free (Temp)
6) Return K
7) Exit
```

Algorithm to delete a node at the end of the list:-



```
Deleteatend(INFO, NEXT, PREV, START)
1) if (START == NULL)
      Write: Underflow and
             Exit
2) if (NEXT[START] == NULL)   // List contain a single
                                 node.
       Set temp = START
       Set START = NULL
   else
    Set temp = START
     Set temp1 = NULL
```

```
while(NEXT[Temp]!=NULL)
    {
     Set Temp1 = temp     //To Keep track of second
                            last node.
    Set Temp = NEXT[Temp]
    }
    Set NEXT[temp1] = NULL
3) Set K = INFO[Temp]
4) free[Temp)
5) Return K
6) Exit
```