Week -7

05/02/2024

WAP to implement a doubly linked list with following operations

① Insertion before a given element
② Deletion of given node

Input →

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
  struct node *next;
  int data;
  struct node *prev;
};

struct node * start = NULL;
struct node * create_ll (struct node *);
struct node * display (struct node *);
struct node * insert_before ( struct node *);
struct node * delete_selected ( struct node *);

int main ()
{
  int option;
do
 {
  pf("\n *Main Menu* ");
  pf("\n 1: Create a list");
  pf("\n 2: Display the list");
  pf("\n 3: Add a node befor an element");
  pf("\n 4: Delete a given node");
  pf("\n 5: exit");
  pf(" enter your option :");
  sf(" %d", & option);
  switch (option)
  {
   case 1:
       start = create_ll (start);
       pf(" \n Doubly_ll created");
       break;
   }
   case 2:
     start = display (start);
      break;
   }
   case 3:
     start = insert_before (start);
     break;
   case 4:
     start = delete_selected (start);
     break;
  }
 } while (option != 5);
   getch();
   return 0;
}
```

```
struct node *create_ll (struct node *start)
{
   struct node *new_node, *ptr;
   int num;
   pf("\n enter -1 to exit ");
   pf ("\n enter the data ");
   sf(" %d", & num);
   while (num != -1)
   {
    if (start == NULL)
    {
     newnode = (struct node *) malloc(sizeof (struct
               node));
     newnode -> prev = NULL;
     newnode -> data = num;
     newnode -> next = NULL;
     start -> newnode;
    }
   else
    {
    ptr = start;
    newnode = (struct node*) malloc (sizeof(struct node));
    newnode => data = num;
    while (ptr -> next != NULL);
          ptr = ptr -> next;
    ptr -> next = newnode;
    newnode -> prev = ptr;
    newnode -> next = NULL;
   }
   pf("enter the data:");
   sf(" %d", & num);
  }
 return
```

```
struct node * inser ~~display~~ (struct node *start)
{ struct node *newnode, *ptr;
  int num, val;
  pf("\n Enter the data:");
  sf("%d", &num);
  pf("\n Enter the value before which the data has to be
        inserted");
  sf("%d", &val);
  newnode = (struct node *) malloc(sizeof(struct node));
    newnode->data = num;
     ptr = start;
   while (ptr->data != val)
         ptr = ptr->next;
   newnode->next = ptr;
   newnode->prev = ptr->prev;
   ~~ptr~~ptr->prev->next = newnode;
    ptr->prev = new_node;
   return start;
}

struct node display (struct node *start){
     struct node *ptr;
     ptr = start;
   while (ptr != NULL)
    { pf("\t %d", ptr->data);
      ptr = ptr->next;
    }
  return start;
}
```

```
struct node delete selected (struct node *start)

{
 struct node *ptr;
  int val;
   ptr = start;
   pf("\n enter the value to be deleted");
   sf("%d", &val);
   while (ptr->data != val)
    {
     ptr = ptr->next;
    }
  if (ptr->data == val)
   {
     ptr->prev->next = ptr->next;
     ptr->next->prev = ptr->prev;
     free(ptr);
   }
 else
   { pf(" Node with %d value doesn't exist \n", val);
 }
 ~~ptr~~
  return start;
}
```

o/p ⇒

```
            Main Menu

        1) Create a list
        2) Display the list
        3) Add a node before a given node
        4) Delete a given node
        5) Exit

Enter your option : 1
Enter -1 to end
 Enter the data : 23   45   6   7   -1
Enter your option : 2

        23   45   6   7

Enter your option : 3
 Enter the data : 6

Enter the value before which the data has to insert
 : 45

Enter your option : 4
Enter the val to be deleted : 6
Enter your option : 2

 23   49   6   45   7
```

```
Enter option : 2
          100, 20, 10, 30, 200, 150, 300.

Enter option : 3
        10, 20, 30, 100; 150, 200, 300

Enter option : 4
        10, 30, 20, 200, 150, 300, 100.
```

[illegible]
19-02-24