LAB - 3

22/1/24

1) WAP to implement singly linked list with following operations.
(a) create a linked list.
(b) Insertion of a node at 1st position, at any position and at end of list.
(c) Display the contents of the linked list.

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<malloc.h>
struct node
{
    int data;
    struct node *next;
};
struct node *start = NULL;
struct node *create_ll(struct node *);
struct node *display(struct node *);
struct node *insert_beg(struct node *);
struct node *insert_end(struct node *);
struct node *insert_before(struct node *);
struct node *insert_after(struct node *);
```

```
int main()
{
    int choice;
    printf("\n *** main menu ***\n 1. create linked list
    \t 2. Display \t 3. Insert_beg \t 4. Insert_end
    \t 5. Insert_before \t 6. Insert_after");
    do
    {
        printf("\n Enter your choice:");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1 : start = create_ll(start);
                     printf("linked list created");
                     break;
            case 2 : start = display(start);
                     break;
            case 3 : start = insert_beg(start);
                     break;
            case 4 : start = insert_end(start);
                     break;
            case 5 : start = insert_before(start);
                     break;
            case 6 : start = insert_after(start);
                     break;
        }
    } while(choice != 7);
```

```
    return 0;
}
struct node *create_ll (struct node *start)
{
    struct node *new_node, *ptr;
    int num;
    printf ("\n Enter -1 to exit");
    printf ("\n Enter the data:");
    scanf ("%d", &num);
    while (num != -1)
    {
        new_node = (struct node*) malloc (sizeof(struct node));
        new_node -> data = num;
        if (start == NULL)
        {
            new_node -> next = NULL;
            start = new_node;
        }
        else
        {
            ptr = start;
            while (ptr -> next != NULL)
                ptr = ptr -> next;
            ptr -> next = new_node;
            new_node -> next = NULL;
        }
```

```
    printf("\n Enter the data : ");
    scanf("%d", &num);
  }
  return start;
};

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

struct node *insert_beg(struct node *start)
{
    struct node *new_node;
    int num;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    new_node=(struct node*)malloc(sizeof(struct node));
    new_node->data = num;
```

```
    new_node -> next = start ;
    return start ;
} ;
struct node *insert_end ( struct node *start )
{
    struct node *new_node , *ptr ;
    int num ;
    printf ( "\n Enter the data : " ) ;
    scanf ( "%d", &num ) ;
    new_node = (struct node*) malloc (sizeof(struct node));
    new_node -> data = num ;
    new_node -> next = NULL ;
    ptr = start ;
    if (start == NULL)
    {
        start = new_node ;
    }
    else
    {
        while (ptr -> next != NULL)
            ptr = ptr -> next ;
        ptr -> next = new_node ;
        return start ;
    }
} ;
```

```
struct node *insert_before(struct node *start)
{
    struct node *new_node, *ptr, *preptr;
    int num, val;
    printf("\n\nEnter the data: ");
    scanf("%d", &num);
    printf("\n Enter the value before which data has to
            be insested: ");
    scanf("%d", &val);
    new_node = (struct node *) malloc(sizeof(struct node));
    new_node -> data = num;
    ptr = start;
    while(ptr -> data != val)
    {
        preptr = ptr;
        ptr = ptr -> next;
    }
    preptr -> next = new_node;
    new_node -> next = ptr;
    return start;
};

Stuct node *create_ll(struct node *staot)
{
    stouct node *new_node, *ptr;
    int num;
```

```
struct node *insert_after(struct node *start)
{
    struct node *new_node, *ptr, *preptr;
    int num, val;
    printf("\n Enter the data : ");
    scanf("%d", &num);
    printf("\n Enter the value after which data has to
            be inserted : ");
    scanf("%d", &val);
    new_node = (struct node *) malloc (sizeof (struct node));
    new_node -> data = num;
    ptr = start;
    preptr = ptr;
    while(preptr -> data != val)
    {
        preptr = ptr;
        ptr = ptr -> next;
    }
    preptr -> next = new_node;
    new_node -> next = ptr;
    return start;
};
```

Output :

```
*** Main menu ***
1. create linked list  2. Display  3. Insert_beg
4. Insert_end  5. Insert_before  6. Insert_after

Enter your choice : 3
Enter the data : 10

Enter your choice : 4
Enter the data : 40

Enter your choice : 5
Enter the value before which data has to be
    inserted : 40
Enter the data : 20

Enter your choice : 6
Enter the data : 30
Enter the value after which data has to be
  inserted : 20
```

2. WAP to implement singly linked list with following operations.

(a) create a linked list.

(b) deletion of 1st element, specified element and last element in the list.

(c) Display the contents of the linked list.

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<malloc.h>
struct node
{
    int data;
    struct node *next;
};
struct node *start = NULL;
struct node *create_ll(struct node*);
struct node *display(struct node*);
struct node *delete_beg(struct node*);
struct node *delete_end(struct node*);
struct node *delete_node(struct node*);
int main()
{  int choice;
   printf("\n*** main menu ***\n 1.create_ll \t
   2. display \t 3. delete_beg \t 4. delete_end \t
   5. delete_node");
```

```
do
{
    printf("\n Enter your choice : ");
    scanf("%d", &choice);
    switch(choice)
    {
        case 1 : start = create_ll(start);
                 printf("\n linked list created");
                 break;
        case 2 : start = display(start);
                 break;
        case 3 : start = delete_beg(start);
                 break;
        case 4 : start = delete_end(start);
                 break;
        case 5 : start = delete_node(start);
                 break;
    }
}while(choice != 6);
return 0;
}
struct node *create_ll(struct node *start)
{
    struct node *new_node, *ptr;
    int num;
```

```
printf("\n Enter -1 to end");
printf("\n Enter the data:");
scanf("%d", &num);
while (num != -1)
{
    new_node = (struct node*) malloc (sizeof(struct node));
    new_node -> data = num;
    if (start == NULL)
    {
        new_node -> next = NULL;
        start = new_node;
    }
    else
    {
        ptr = start;
        while (ptr -> next != NULL)
            ptr = ptr -> next;
        ptr -> next = new_node;
        new_node -> next = NULL;
    }
    printf("\n Enter the data:");
    scanf("%d", &num);
}
return start;
};
```

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

struct node * delete_beg (struct node * start)
 {
    struct node *ptr ;
    ptr = start ;
    start = start->next ;
    free(ptr);
    return start ;
 };

struct node *delete_end (struct node * start)
{
   struct node *ptr, *preptr;
   ptr = start ;
```

```
while ( ptr -> next != NULL)
{
    preptr = ptr ;
    ptr = ptr->next ;
}
preptr->next = NULL ;
free(ptr) ;
return start ;
};

struct node *delete_node( struct node *start)
{
  struct node *ptr, *preptr ;
  int val ;
  Printf("\n Enter the value to be deleted : ");
  Scanf("%d", &val) ;
  ptr = start ;
  if(ptr->data == val)
  {
     start = delete_beg(start) ;
     return start ;
  }
  else {
     while (ptr->data != val)
     {
        preptr = ptr ;
        ptr = ptr->next ;
     }
```

```
        preptr->next = ptr->next ;
        free(ptr)
        display(start) ;
};
```

output :

```
*** main menu ***
1. create linked list  2. display  3. delete_beg
4. delete_end  5. delete_node

Enter your choice : 1
Enter -1 to end
Enter the data : 10
Enter the data : 20
Enter the data : 30
Enter the data : 40
Enter the data : -1
Enter your choice : 2
10 20 30 40
Enter your choice : 3
Enter your choice : 4
Enter your choice : 2
20 30
Enter your choice : 5
Enter value to be deleted : 20
```

24.01.24