22-01-24

Q) WAP to implement singly linked list with following operations.
(a) create a linked list
(b) insertion of a node at first, at any and at end of list.
Display the contents of linked list

I/p →

```
#include <stdio.h>
#include <malloc.h>

struct node
{
  int data;
  struct node * next;
};

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("Enter the data: ");
scanf (" %d ", &num);
}
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));
```

```
    start = new_node;
    return start;
}

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

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

struct node * start = NULL;
struct node * create_ll (struct node *);
struct node * insert_beg (struct node *);
struct node * insert_end (struct node *);
struct node * insert_before (struct node *);
struct node * display (struct node *);
```

(5) WAP to implement singly linked list with (a) create (b) deletion of first, random element and last element. and display the content.

→
```
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
{
  struct node *ptr, *preptr;
  int val;
  printf("Enter the value after which the
          node has to be deleted:");
  scanf("%d", &val);
  ptr = start;
  preptr = ptr;
  while (preptr -> data != val)
  {
     preptr = ptr;
     ptr = ptr -> next;
  }
  preptr -> next = ptr -> next;
  free(ptr);
  return start;
};
```

```
struct node * delete_beg (struct node *);
struct node * delete_end (struct node *);
struct node * delete_after (struct node *);
struct node * display (struct node *);
struct node * display (struct node *start);
{
  struct node *ptr;
  ptr = start;
```

o/p:- 1. create list 2. Display.
3. insert_beg 4. Insert_end
5. insert_random

Enter choice : 1
Enter -1 to end :
Enter data : 3 4 5 6
enter choice : 2
3 4 5 6
Enter choice : 3
enter value : 2.
enter choice : 2
2 3 4 5 6

Enter choice : 4
Enter value : 7
Enter choice : 2
2 3 4 5 6 7

Enter choice : 5.
Enter value : 9
Enter value before which has to be inserted : 5
~~Enter value~~
Enter choice : 2
2 3 4 5 9 6 7.

22/01/24