Name: Gaurav singh

Roll No: 70 Semester: 3 Sem.

Subject: Data structures and Algorithms.

Practical: 4 To Implement Linked List with Various operations like Create, Insert, Search, Find length.

CODE:

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



void lastinsert ();
void random_delete();
void display();
void search();
void reverseLL();
void main ()
{
   int choice =0;
   while(choice != 9)
   {
      printf("\n\n*********Main Menu*********\n");
      printf("\nChoose one option from the following list ...\n");
      printf("\n===============================================\n");
```

```
        printf("\n1.Insert Element\n2.Delete Element\n3.Display Elements\n4.Search Element\n5.Reverse
the list.\n6.Exit\n");
        printf("\nEnter your choice?\n");
        scanf("\n%d",&choice);
        switch(choice)
        {
           case 1:
           lastinsert();
           break;
           case 2:
           random_delete();
           break;
           case 3:
           display();
           break;
           case 4:
           search();
           break;
           case 5:
           reverseLL();
           break;
           case 6:
           default:
           printf("Please enter valid choice..");
        }
     }
}

void lastinsert()
```

```
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node*)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nEnter value?\n");
        scanf("%d",&item);
        ptr->data = item;
        if(head == NULL)
        {
            ptr -> next = NULL;
            head = ptr;
            printf("\nNode inserted");
        }
        else
        {
            temp = head;
            while (temp -> next != NULL)
            {
                temp = temp -> next;
            }
            temp->next = ptr;
            ptr->next = NULL;
            printf("\nNode inserted");
```

```
        }
    }
}

void random_delete()
{
    struct node *ptr,*ptr1;
    int loc,i;
    printf("\n Enter the location of the node after which you want to perform deletion \n");
    scanf("%d",&loc);
    ptr=head;
    for(i=0;i<loc;i++)
    {
        ptr1 = ptr;
        ptr = ptr->next;

        if(ptr == NULL)
        {
            printf("\nCan't delete");
            return;
        }
    }
    ptr1 ->next = ptr ->next;
    free(ptr);
    printf("\nDeleted node %d ",loc+1);
}
```

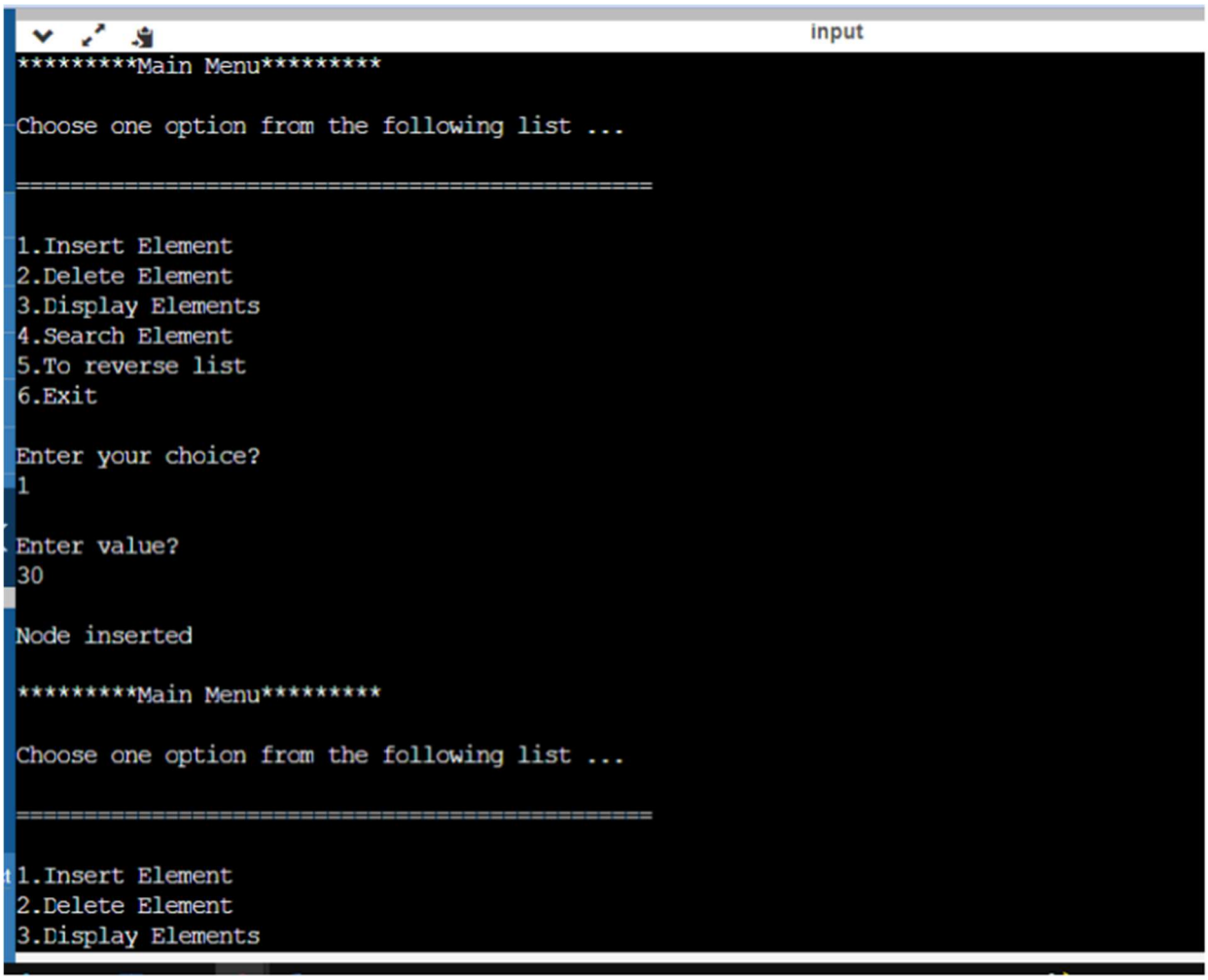
```
void search()
{
   struct node *ptr;
   int  item,i=0,flag;
   ptr = head;
   if(ptr == NULL)
   {
      printf("\nEmpty List\n");
   }
   else
   {
      printf("\nEnter item which you want to search?\n");
      scanf("%d",&item);
      while (ptr!=NULL)
      {
         if(ptr->data == item)
         {
            printf("item found at location %d ",i+1);
            flag=0;
         }
         else
         {
            flag=1;
         }
         i++;
         ptr = ptr -> next;
      }
      if(flag==1)
```

```
        {
            printf("Item not found\n");
        }
    }

}

void display()
{
    struct node *ptr;
    ptr = head;
    if(ptr == NULL)
    {
        printf("Nothing to print");
    }
    else
    {
        printf("\nprinting values.....\n");
        while (ptr!=NULL)
        {
            printf("\n%d",ptr->data);
            ptr = ptr -> next;
        }
    }
}

// Function to reverse the linked list
void reverseLL()
{
```

```
struct node *t1, *t2, *temp, *ptr;
t1 = t2 = NULL;

// If LL is empty
if (head == NULL)
   printf("List is empty\n");

// Else
else {

   // Traverse the LL
   while (head != NULL) {

      // reversing of points
      t2 = head->ptr;
      head->link = t1;
      t1 = start;
      start = t2;
   }
   start = t1;

   // New head Node
   temp = head;

   printf("Reversed linked "
         "list is : ");

   // Print the LL
   while (temp != NULL) {
```

```
            printf("%d ", temp->info);
            temp = temp->link;
        }
    }
}
```

Output:


input
*********Main Menu*********
Choose one option from the following list ...
==========
1.Insert Element
2.Delete Element
3.Display Elements
4.Search Element
5.To reverse list
6.Exit
Enter your choice?
1
Enter value?
30
Node inserted
*********Main Menu*********
Choose one option from the following list ...
==========
1.Insert Element
2.Delete Element
3.Display Elements