Programs:

DATE

1. Write a program to insert data at different positions into a single linked list and count the no. of nodes present in the single linked list.

```
# include <stdio.h>
# include <conio.h>
# include <malloc.h>
 struct node
     {   int info;
         struct node * link;
     }*start;

main()
   {   int choice, n, m, position, i;
       clrscr();
       start = NULL;
       while ( choice != 6)
       {   printf ( " 1. Create List \n");
           printf (" 2. Add at beginning");
           printf (" 3. Add after");
           printf ("4. Display");
           printf ("5. Count");
           printf ("6. Quit");
           printf (" Enter your choice :");
           scanf ("%d", &choice);
           switch (choice)
           { case 1 : printf (" How many nodes you want:")
                      scanf ("%d", &n);
                      for(i=0; i<n; i++)
                        { printf (" Enter the element :");
                          scanf ("%d", &m);
                          create_list(m);
                        }
```

```
            break;
case 2:     printf("Enter the element: ");
            scanf("%d", &m);
            addatbeg(m);
            break;
case 3:     printf("Enter the element: ");
            scanf("%d", &m);
            printf("Enter The position after which this element is
                    inserted :");
            scanf("%d", &position);
            addafter(m, position);
            break;
case 4:     Display();
            break;
case 5:     Count();
            break;
case 6:     exit();
default:    printf("Wrong choice\n");
        }
    }
}
create_list(int data)
{   struct node *q, *temp;
    temp = malloc(sizeof(struct node));
    temp->info = data;
    temp->link = NULL;
    if(start == NULL)
        start = temp;
    else
    {   q = start;
        while(q->link != NULL)
            q = q->link;
        q->link = temp;
    }
}
```

```
addatbeg (int data)
    {   struct node *temp;
        temp= malloc (size of (struct node));
        temp->info = data;
        temp->link = start;
        start = temp;
    }
addafter (int data, int pos)
    {
        struct node *temp, *q;
        int i;
        q = start;
        for (i=0; i<pos-1; i++)
            {
                q = q->link;
                if (q==NULL)
                { printf ("There are less than %d elements", pos);
                return;
                }
            }
        temp= malloc (size of struct node));
        temp->link = q->link;
        temp->info = data;
        q->link    = temp;
    }
```

```
temp->info = data;
q->link->prev = temp;
temp->link = q->link;
temp->prev = q;
q->link = temp;
```

```
display()
{
    struct node *q;
    if (start == NULL)
    { printf ("list is empty \n");
      return;
    }
    q = start;
    printf ("list is : \n");
    while (q != NULL)
    { printf ("%d", q->info)
      q = q->link;
    }
    printf ("\n");
}
count().
{ struct node *q = start;
  int cnt = 0;
  while (q != NULL)
  { q = q->link;
    cnt++;
  }
  printf ("Number of elements are %d\n", cnt);
}
```

DATE

Q ② Write a program to delete data which are at different positions of a linked list.

```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
 struct node
     {   int info;
         struct node *link;
     } *start;
main()
 {
    int choice, n, m, position, i;
    start = NULL;
    while (choice != 4)
      { printf ("1. Create List \n");
        printf ("2. Delete \n");
        printf ("3. Display\n");
        printf ("4. Quit\n");
        printf ("Enter your choice:");
        scanf ("%d", &choice);
        switch (choice)
        { case 1: printf ("How many nodes you want: ");
                  scanf ("%d", &n);
                  for(i=0; i<n; i++)
                  {  printf ("Enter the element:");
                     scanf ("%d", &m);
                     create_list(m);
                  }
                  break;
```

```
case 2 : if (start == NULL)
          { printf ("List is empty \n");
            continue;
          }
          printf ("Enter the element for deletion:");
          scanf ("%d", &m);
          del(m);
          break;
case 5 : Display();
          break;
case 6 : exit();
default : printf ("Wrong choice \n");
    }
  }
}
create_list (int data)
{   struct node *q, *temp;
    temp = malloc (sizeof(struct node));
    temp -> info = data;
    temp -> link = NULL;
    if (start == NULL)
          start = temp;
    else
    {   q = start;
        while (q -> link != NULL)
              q = q -> link;
        q -> link = temp;
    }
}
```

```
del(int data)
{
    struct node *temp, *q;
    if (start->info == data)
    {
        temp = start;
        start = start->link;
        free(temp);
        return;
    }
    q = start;
    while (q->link->link != NULL)
    {
        if (q->link->info == data)
        {
            temp = q->link;
            q->link = temp->link;
            free(temp);
            return;
        }
        q = q->link;
    }
    if (q->link->info == data)
    {
        temp = q->link;
        free(temp);
        q->link = NULL;
        return;
    }
    printf("Element %d not found\n", data);
}
```

Display ()

```
{   struct node *q;
    if (start == NULL)
    {   printf ("List is empty\n");
        return;
    }
    q = start;
    printf("List is :\n");
    while ( q != NULL)
      {
          printf ("%d", q->info);
          q = q->link;
      }
     printf("\n");
}
```

DATE

Q 3. Write a program to reverse a linked list.

```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
struct node
    {   int info;
        struct node *link;
    }*start;
main()
{
    int i, n, item;
    start = NULL;
    printf("How many nodes you want: ");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {   printf("Enter the item:%d", i+1);
        scanf("%d", &item);
        create_list(item);
    }
    printf("Initially the linked list is:\n");
    display();
    reverse();
    printf("Linked list after reversing is:\n");
    display();
}
create_list(int data)
{   struct node *q, *temp;
    temp = malloc(sizeof(struct node));
    temp->info = data;
    temp->link = NULL;
```

```
    if (start == NULL)
            start = temp;
    else
    {       q = start;
            while (q->link != NULL)
                q = q->link;
            q->link = temp;
    }
}
display()
{
        struct node *q;
        if (start == NULL)
        {   printf ("List is empty\n");
            return;
        }
        q = start;
        while (q != NULL)
        {   printf ("%d", q->info);
            q = q->link;
        }
        printf ("\n");
}
reverse()
{   struct node *p1, *p2, *p3;
        if (start->link == NULL)
            return;
        p1 = start;
        p2 = p1->link;
        p3 = p2->link;
        p1->link = NULL;
        p2->link = p1;
```

```
while (p3 != NULL)
{
    p1 = p2 ;
    p2 = p3 ;
    p3 = p3 -> link ;
    p2 -> link = p1 ;
}
start = p2 ;
}
```