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


                  void create();
                  void display();
                  void insert_beg();
                  void insert_end();
                  void insert_pos();


                  struct Node
                  {
                     int data;
                     struct Node *link;
                  };
                  typedef struct Node node;
                  node *start=NULL;


                  void main()
                  {
                     int ch,ch1;
                     while(1)
                     {
                         printf("1.Create 2.Insert 3.Display 4.Exit\n");
                         printf("Enter your choice:\n");
                         scanf("%d",&ch);
                         switch(ch)
                         {
                         case 1:
```

```
            create();
            break;
        case 2:
            printf("Insert at 1.Beginning 2.At any position
3.End\n");
            scanf("%d",&ch1);
            if(ch1==1)
                insert_beg();
            else if(ch1==2)
                insert_pos();
            else if (ch1==3)
                insert_end();
            else
                printf("Invalid choice\n");


            break;


        case 3:
            display();
            break;
        case 4:
            exit(1);
        default:
            printf("Invalid choice\n");
        }
    }
}


void create()
{
    int c;
```

```
   node *new,*curr;
   start=(node *) malloc(sizeof(node));
   curr=start;
   printf("Enter element\n");
   scanf("%d",&start->data);
   while(1)
   {
       printf("Do you want to add another element(1 for Yes / 0
for No)\n");
       scanf("%d",&c);
       if(c==1)
       {
           new=(node *) malloc(sizeof(node));
           printf("Enter element\n");
           scanf("%d",&new->data);
           curr->link = new;
           curr=new;
       }
       else
       {
           curr->link=NULL;
           break;
       }
   }
}


void display()
{
   node *temp;
   if(start==NULL)
   {
       printf("Linked list is empty\n");
```

```
        return;
    }
    temp=start;
    while(temp!=NULL)
    {
        printf("%d\n",temp->data);
        temp = temp->link;
    }
}

void insert_beg()
{
    node *new;
#include<stdio
.h>
#include<stdlib.h>
#include<malloc.h>

void create();
void display();
void insert_beg();
void insert_end();
void insert_pos();

struct Node
{
    int data;
    struct Node *link;
};
typedef struct Node node;
node *start=NULL;
```

```
void main()
{
   int ch,ch1;
   while(1)
   {
       printf("1.Create 2.Insert 3.Display 4.Exit\n");
       printf("Enter your choice:\n");
       scanf("%d",&ch);
       switch(ch)
       {
       case 1:
           create();
           break;
       case 2:
           printf("Insert at 1.Beginning 2.At any position
3.End\n");
           scanf("%d",&ch1);
           if(ch1==1)
               insert_beg();
           else if(ch1==2)
               insert_pos();
           else if (ch1==3)
               insert_end();
           else
               printf("Invalid choice\n");


           break;


       case 3:
           display();
```

```
            break;
        case 4:
            exit(1);
        default:
            printf("Invalid choice\n");
        }
    }
}


void create()
{
    int c;
    node *new,*curr;
    start=(node *) malloc(sizeof(node));
    curr=start;
    printf("Enter element\n");
    scanf("%d",&start->data);
    while(1)
    {
        printf("Do you want to add another element(1 for Yes / 0
for No)\n");
        scanf("%d",&c);
        if(c==1)
        {
            new=(node *) malloc(sizeof(node));
            printf("Enter element\n");
            scanf("%d",&new->data);
            curr->link = new;
            curr=new;
        }
        else
        {
```

```
            curr->link=NULL;
            break;
        }
    }
}

void display()
{
    node *temp;
    if(start==NULL)
    {
        printf("Linked list is empty\n");
        return;
    }
    temp=start;
    while(temp!=NULL)
    {
        printf("%d\n",temp->data);
        temp = temp->link;
    }
}

void insert_beg()
{
    node *new;
    new=(node *) malloc(sizeof(node));
    printf("Enter element\n");
    scanf("%d",&new->data);
    if(start==NULL)
    {
        start=new;
```

```
        new->link=NULL;
        return;
    }
    new->link=start;
    start=new;
}

void insert_end()
{
    node *new,*temp;
    new=(node *)malloc(sizeof(node));
    printf("Enter element\n");
    scanf("%d",&new->data);
    if(start==NULL)
    {
        start=new;
        new->link=NULL;
        return;
    }
    temp=start;
    while(temp->link!=NULL)
    {
        temp=temp->link;
    }
    temp->link=new;
    new->link=NULL;
}

void insert_pos()
{
    int pos;
```

```
    node *new,*temp;
    new=(node *)malloc(sizeof(node));
    printf("Enter element\n");
    scanf("%d",&new->data);
    printf("Enter position\n");
    scanf("%d",&pos);
    if(pos==1)
    {
        new->link=start;
        start=new;
        return;
    }
    int i=1;
    temp=start;
    while(i<pos-1 && temp->link!=NULL)
    {
        temp=temp->link;
        i++;
    }
    if(i==(pos-1))
    {
        new->link=temp->link;
        temp->link=new;
        return;
    }
    if(temp==NULL)
    {
        printf("Invalid position");
    }
}
```