~~import~~

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

void delete()
{
    struct node* ptr, struct node * temp;
    int val;
    printf(" Enter the value:");
    scanf(" %d ", &val);
     temp=head;
     while (temp -> data != val)
          temp = temp -> next;
      if (temp -> next == NULL)
          printf(" cannot delete");
      else if (temp -> next -> next == NULL)
          temp -> next = NULL;
          printf(" last Node deleted");
      else
      {
           ptr = temp -> next;
           temp -> next = ptr -> next;
           ptr -> next -> prev = temp;
            free(ptr);
            printf(" Node deleted");
      }
}
```

```
void display() {
    struct node* ptr = head;

    if (ptr == head)
        printf("Empty");
    else
    {
        printf("\n\n LIST ->");
        while (ptr != NULL)
        {
            printf("%d", ptr->data);
            ptr = ptr->next;
        }
    }
}

void create_list() {

    struct node *ptr;
    int i, n, new_data;
    printf("Enter the no of nodes");
    scanf("%d", &n);

    if (n >= 1)
    {
        head = (struct node *) malloc (sizeof(struct node));
        if (head != NULL)
        {
            printf("Enter value for Node 1 ");
            scanf("%d", &new_data);
```

```
head-> data = new_data;
head-> prev = NULL;
head-> next = NULL;
last = head;
for (i = 2; i <= n; i++)
{
  ptr = (struct node *) malloc (sizeof (struct node));
  if (ptr != NULL)
  {
    printf("Enter value for Node %d: ", i)
    scanf("%d", &new_data);
    ptr -> data = new_data;
    ptr -> prev = last;
    ptr -> next = NULL;
    last -> next = ptr;
    last = ptr;
  }
  printf("Linked list created");
}
else
{
  printf("Nodes cannot be created");
}
}
}

void insert() {
  int i, position, new_data;
  struct node * ptr, * temp;
  if (head == NULL)
  {
    printf("List is empty");
```

```
else
{
    temp = head;
    i = 1;
    while (i < position - 1 && temp != NULL)
    {
        temp = temp -> next;
        i++;
    }
    if (position == 1)
    {
        ptr -> data = new_data;
        ptr -> next = head;
        ptr -> prev = ptr;
        head = ptr;
    }
    else if (temp == last)
    {
        ptr -> data = new_data;
        ptr -> next = NULL;
        ptr -> prev = last;
        head -> prev = ptr;
        head = pt
        last -> next = ptr;
        last = ptr;
    }
    else if (temp != NULL)
    {
        ptr -> data = new_data;
        ptr -> next = temp -> next;
        ptr -> prev = temp;
```

```
if (temp->next != NULL)
{
    temp->next->prev = ptr;
}
    temp->next = ptr;
}
else
{
    printf(" Invalid position");
}
}
```