```c
#include<stdio.h>
#include<stdlib.h>
struct student
{
      int s_no;
      char s_name[20];
      struct student *ptr;
};
struct student *create_list(struct student *start)
{
      char ch;
      struct student *s1=0,*dummy=0;
      do
      {
            dummy=(struct student*)malloc(sizeof(struct student));
            printf("Enter student number & name:");
            scanf("%d%s",&dummy->s_no,dummy->s_name);
            if(start==NULL)
                  start=dummy;
            else
                  s1->ptr=dummy;
                  s1=dummy;
            printf("Do you want to coninuee:(y/n)");
            scanf(" %c",&ch);
      }while(ch=='y');
      dummy->ptr=NULL;
      return start;
}
struct student *insert_before(struct student *start)
{
      if(start!=NULL)
      {
            int ele;
            struct student *i=0,*nn,*prev=0;
            printf("Which element before do you want to insert:");
            scanf("%d",&ele);
            nn=(struct student*)malloc(sizeof(struct student));
            printf("Enter no and name for insert:");
            scanf("%d%s",&nn->s_no,nn->s_name);
            if(start->s_no==ele)
            {
                  nn->ptr=start;
                  start=nn;
            }
            else
            {
                  for(i=start,prev=i;i!=NULL;prev=i,i=i->ptr)
                  {
                        if(i->s_no==ele)
                        {
                              nn->ptr=i;
                              prev->ptr=nn;
                              break;
                        }
                  }
            }
      }
      else
            printf("\nNo Elements in the List to insert\n");
      return start;
}
struct student *insert_after(struct student *start)
{
      if(start!=NULL)
      {
            int ele;
            struct student *i=0,*nn;
            printf("Which element after do you want to insert:");
            scanf("%d",&ele);
            nn=(struct student*)malloc(sizeof(struct student));
            printf("Which element do you want to insert:");
            scanf("%d%s",&nn->s_no,nn->s_name);
            if(start->ptr==NULL)
                  start->ptr=nn;
            else
            {
                  for(i=start;i!=NULL;i=i->ptr)
                  {
                        if(i->s_no==ele)
                        {
                              nn->ptr=i->ptr;
                              i->ptr=nn;
                              break;
                        }
                  }
            }
      }
      else
            printf("\nNo Elements in the List to insert\n");
      return start;
}
struct student *delete_before(struct student *start)
{
      int ele;
      if(start!=NULL)
      {
            struct student *i=0,*prev=0,*pprev=0;
            printf("Enter element which is after u want to delete:");
            scanf("%d",&ele);
            if(start->s_no==ele)
                  printf("Sorry! No elements in the before given element\n");
            else if(start->ptr!=NULL)
            {
                  for(i=start,prev=start,pprev=prev;i!=NULL;pprev=prev,prev=i,i=i->ptr)
                  {
                        if(i->s_no==ele)
                        {
                              pprev->ptr=i;
                              printf("Before node has deleted\n");
                              break;
                        }
                  }
            }
      }
      else
            printf("\nNo Elements in the List to delete\n");
      return start;
}
struct student *delete_after(struct student *start)
{
      int ele;
      if(start!=NULL)
      {
            struct student *i=0;
            printf("Enter element which is before to element,u want to delete:");
            scanf("%d",&ele);
            if(start->ptr==NULL)
                  printf("Sorry! No elements in the after given element\n");
            else
            {
                  for(i=start;i!=NULL;i=i->ptr)
                  {
                        if(i->s_no==ele)
                              i->ptr=(i->ptr)->ptr;
                  }
            }
      }
      else
            printf("\nNo Elements in the List to delete\n");
      return start;
}
struct student *update_list(struct student *start)
{
      if(start!=NULL)
      {
            int ele;
            struct student *i;
            printf("Enter element no. which you want to update:");
            scanf("%d",&ele);
            if(start==NULL)
                  printf("Sorry! no element in the list\n");
            else
            {
                  for(i=start;i!=NULL;i=i->ptr)
                  {
                        if(i->s_no==ele)
                        {
                              printf("What is updated no. and name:");
                              scanf("%d%s",&i->s_no,i->s_name);
                              break;
                        }
                  }
            }
      }
      else
            printf("\nNo Elements in the List to delete\n");
      return start;
}
struct student *display_list(struct student *start)
{
      struct student *i;
      if(start==NULL)
            printf("Sorry! No elements in the list\n");
      else
      {
            printf("\tNumber\tname\n");
            for(i=start;i!=NULL;i=i->ptr)
                  printf("\n\t%d\t%s",i->s_no,i->s_name);
            printf("\n\n");
      }
}
main()
{
      system("clear");
      struct student *start=NULL;
      int ch;
      do
      {
            printf("\t\tMENU\n");
            printf("\t1->CREATE LIST\n");
            printf("\t2->INSERT BEFORE\n");
            printf("\t3->INSERT AFTER\n");
            printf("\t4->DELETER BEFORE\n");
            printf("\t5->DELETE AFTER\n");
            printf("\t6->UPDATE THE PARTICULAR\n");
            printf("\t7->DISPLAY THE LIST\n");
            printf("\t8->EXIT\n");
            printf("Enter your choice:(1-8)");
            scanf("%d",&ch);
            switch(ch)
            {
```

```
            case 1:
                  start=create_list(start);
                  break;
            case 2:
                  start=insert_before(start);
                  break;
            case 3:
                  start=insert_after(start);
                  break;
            case 4:
                  start=delete_before(start);
                  break;
            case 5:
                  start=delete_after(start);
                  break;
            case 6:
                  start=update_list(start);
                  break;
            case 7:
                  display_list(start);
                  break;
            case 8:
                  printf("Thank You!!!\n");
                  break;
            default:
                  printf("INVALID OPTION\n");
            }
      }while(ch!=8);
}
```