Code for Singly Linked List

```
#include<conio.h>
#include<string.h>
#include<stdlib.h>
# include<iostream.h>
# define SIZE 1
struct student
{
     char *name;
     int roll_no;
     int marks[SIZE];
     struct student *ptr_next;
}*ptr_this,*ptr_first;
void print_menu(void);
void add_record(void);
void delete_record(void);
void search_record(void);
void sort_records(void);
void display_records(void);
void main (void)
{
     char choice;
     ptr_this = ptr_first = (struct student*)NULL;
     while(1)
```

```
{
    print_menu();
    choice = getche();
    switch(choice)
    {
     case '1': //add record
         add_record();
         break;
     case '2': //delete record
         delete_record();
         break;
     case '3': //search record
         search_record();
         break;
     case '4': //sort record
         sort_records();
         break;
     case '5': //display records
         display_records();
         break;
     case '6': //exit
         exit(0);
     default:
         cout<<"\n\n\tEnter valid choice number";
```

```
                getch();
            }
        }
}
void print_menu(void)
{
    clrscr();
    cout<<"\n\n\t\t\tDATABASE SUBSYSTEM";
    cout<<"\n1. Add record";
    cout<<"\n2. Delete record";
    cout<<"\n3. Search record";
    cout<<"\n4. Sort records";
    cout<<"\n5. Display records";
    cout<<"\n6. Exit";
    cout<<"\n\n\tEnter your choice number: ";
    }
void add_record(void)
{
    char temp[80];
    int i;
    struct student *ptr_new;
    ptr_new = new student;
    if(ptr_new == (struct student*)NULL)
    {
```

```
        cout<<"\n\n Not enough memory";
        return;
    }
    cout<<"\n\nEnter Name: ";
    cin>>temp;
    ptr_new->name=new char [strlen(temp)+1];
    strcpy(ptr_new->name,temp);
    cout<<"Enter Roll No.: ";
    cin>>ptr_new->roll_no;
    cout<<"Enter Marks: \n";
    for(i=0;i<SIZE;i++)
    {
        cout<<"Course # "<< i+1 << " : ";
        cin>> ptr_new->marks[i];
    }
    ptr_new->ptr_next = (struct student*)NULL;

    if(ptr_first == (struct student*)NULL)
        ptr_this = ptr_first = ptr_new;
    else
        {
            ptr_this-> ptr_next = ptr_new;
            ptr_this = ptr_new;
         }
```

```
     cout<<"\n\n\t Record has been added";
     getch();
}
void delete_record(void)
{
     char temp[10];
     int rec_no,j=0;
     struct student *ptr_general, *ptr_prev;
     cout<<"\n\n Enter record number to be deleted: ";
     cin>>rec_no;
     ptr_general = ptr_prev= ptr_first;
     while(ptr_general != (struct student*)NULL)
      {
          j++;
          if(j == rec_no)
          {
           if(ptr_general == ptr_first)
               ptr_first = ptr_first->ptr_next;
           else
               ptr_prev->ptr_next = ptr_general->ptr_next;
           free(ptr_general);
           cout<<"\n\nRecord # "<< rec_no <<"has been
deleted";
           getch();
```

```
            return;
           }
           ptr_prev = ptr_general;
           ptr_general = ptr_general->ptr_next;
       }
}
void sort_records(void)
{
     struct student *count_out, *count_in, *dummy;
     count_out = ptr_first;
     while(count_out->ptr_next != (struct student *)NULL)
     {
          count_in = count_out->ptr_next;
          while(count_in != (struct student*)NULL)
          {
           if(strcmp(count_out->name,count_in->name)>0)
           {
               *dummy = *count_out;
               *count_out = *count_in;
               *count_in = *dummy;
               //interchanging ptr_next
               dummy->ptr_next = count_out->ptr_next;
               count_out->ptr_next = count_in->ptr_next;
               count_in->ptr_next = dummy->ptr_next;
```

```
            }
            count_in = count_in->ptr_next;
           }
          count_out = count_out->ptr_next;
     }
     cout<<"\n\n\t Records has been sorted in ascending
order of names";
     getch();
}
void search_record(void)
{
     //char temp[10];
     int i,roll,flag=0;
     struct student *ptr_general;
     cout<<"\n\n \t Enter roll no. to be searched: ";
     cin>>roll;
     ptr_general = ptr_first;
     while(ptr_general != (struct student*)NULL)
     {
          if(ptr_general->roll_no == roll)
          {
           cout<<"\n\n\nName = "<<ptr_general->name;
           cout<<"\nRoll No. = "<<ptr_general->roll_no;
           for(i=0;i<SIZE;i++)
```

```
                cout<<"\nMarks for course # "<<i+1 <<" = "<<
ptr_general->marks[i];
           flag=1;
           getch();
          }
          ptr_general = ptr_general->ptr_next;
     }
     if(flag == 0)
          cout<<"\n\nRecord not found";
     else
          cout<<"\n\nNo more records";
     getch();
}
void display_records(void)
{
     struct student *ptr_general;
     int i,j=1;
     char ch;
     ptr_general = ptr_first;
     do
     {
          if(ptr_general != (struct student*)NULL)
          {
           cout<<"\n\nRecord # " << j++;
```

```
          cout<<"\nName = "<< ptr_general->name;
          cout<<"\nRoll No. = "<<ptr_general->roll_no;
          for(i=0;i<SIZE;i++)
              cout<<"\nMarks for course # "<<i+1<< " =
"<<ptr_general->marks[i];
          ptr_general = ptr_general->ptr_next;
          cout<<"\n\npress ESC to exit or ENTER to see the
next record: ";
          ch = getch();
          if(ch == 27)
              return;
         }
         else
         {
          cout<<"\n\nNo more records";
          getch();
          return;
         }
     }while(ch == 13);
}
```
