www.nsgacademy.in
NET SECRETS GROUP, Pinnacle Pride, 1st Floor,
Above Maharashtra Electronics, Near Durvankur Dining Hall,
Opposite Cosmos Bank, Tilak Road, Sadashiv Peth, Pune-411030
Contact No: 9823782121 / 020 65000223

LINE EDITOR

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

typedef struct node
{
 char line[80];
 struct node* next;
}node;

char fname[20];
FILE* fp;
int cnt,c;
node *first,*last,*first1,*last1;

void create()
{
 char str[80];
 node* temp;

 while(!feof(fp))
      {
      fgets(str,80,fp);

      temp=(node*)malloc(sizeof(node));
      strcpy(temp->line,str);
      temp->next=NULL;

      if(first==NULL)
         first= temp;
      else
         last->next=temp;

      last=temp;
      cnt++;
      }
}
```



```
void createnew()
{
 char str[80];
 node* temp;

 printf("Enter text (. to stop)\n");
 gets(str);

 while(strcmp(str,".")!=0)
       {
        temp=(node*)malloc(sizeof(node));
        strcpy(temp->line,str);
        temp->next=NULL;

        if(first==NULL)
            first = temp;
        else
            last->next = temp;

        last = temp;
        cnt++;

        gets(str);
       }
}

void save()
{
 node* p;
 fp=fopen(fname,"w");

 p=first;
 while(p!=NULL)
      {
       fputs(p->line,fp);
       p=p->next;
      }
 printf("%s saved successfully...\n",fname);

 fclose(fp);
}
```



```
void disppattern(char *str)
{
 node* p;
 int i=1;
 p=first;
 while(p!=NULL)
      {
       if(strstr(p->line,str))
          printf("%d: %s",i,p->line);

       p=p->next;
       i++;
      }
}

node* findnode(int pos)
{
 node* p;
 int i=1;

 p=first;
 while(p!=NULL && i<pos)
      {
       p=p->next;
       i++;
      }
 return p;
}

void printnodes(int m,int n)
{
 node* p;
 int i=0;

 p=findnode(m);
 while(p!=NULL && i<=n-m)
      {
       printf("%d: %s",i+m,p->line);
       i++;
       p=p->next;
      }
 printf("\n");
}
```



```
void insertnodes(int no)
{
 node *temp,*p;
 char str[80];

 first1=last1=NULL;
 c=0;

 printf("Enter text (. to stop)\n");
 gets(str);

 while(strcmp(str,".")!=0)
      {
       temp = (node*)malloc(sizeof(node));
       strcpy(temp->line,str);
       temp->next=NULL;

       if(first1==NULL)
         first1 = temp;
       else
         last1->next = temp;

       last1 = temp;
       c++;

       gets(str);
      }

 if(no==1)
   {
    last1->next=first;
    first=first1;
   }
 else
   {
    p=findnode(no-1);

    last1->next=p->next;
    p->next=first1;
   }

 cnt+=c;
}
```



```
void copynodes(int x,int y)
{
 node *p,*temp;
 c=0;
 first1=last1=NULL;

 p=findnode(x);

 while(p!=NULL && y>=x)
       {
        temp=(node*)malloc(sizeof(node));
        strcpy(temp->line,p->line);
        temp->next=NULL;

        if(first1==NULL)
          first1=temp;
        else
          last1->next=temp;

        last1=temp;
        p=p->next;
        y--;
        c++;
      }
}

void pastenodes(int z)
{
 node* p;

 p=findnode(z-1);

 last1->next=p->next;
 p->next=first1;

 cnt+=c;
}
```



```
void deletenodes(int m,int n)
{
 node *temp,*p;

 if(m==1)
   {
    while(n>0)
         {
          temp=first;
          first=temp->next;
          temp->next=NULL;
          free(temp);
          cnt--;
          n--;
         }
   }
 else
    {
     p=findnode(m-1);

     while(n>=m)
            {
             temp=p->next;
             p->next=temp->next;
             temp->next=NULL;
             free(temp);
             cnt--;
             n--;
            }
    }
}

void movenodes(int x,int y,int z)
{
 copynodes(x,y);
 pastenodes(z);
 deletenodes(x,y);
}
```



```
void main(int argc, char* argv[])
{
 char cmd[20],tok1[5],tok2[5],tok3[5],tok4[5];
 int n,t2,t3,t4;

 strcpy(fname,argv[1]);

 fp=fopen(fname,"r");
 if(fp==NULL)
   {
    printf("File doesn't exist\n");
    createnew();
    printf("\nNumer of lines is %d",cnt);
   }
 else
   {
    printf("\nFile exist");
    create();
    printf("\nNumer of lines is %d",cnt);
    fclose(fp);
   }

 while(1)
      {
       strcpy(tok2,"\0");
       strcpy(tok3,"\0");
       strcpy(tok4,"\0");

       printf("\n\n$ ");

       gets(cmd);

       n=sscanf(cmd,"%s %s %s %s",tok1,tok2,tok3,tok4);

       t2=atoi(tok2);
       t3=atoi(tok3);
       t4=atoi(tok4);
```



```
        switch(tok1[0])
             {
             case 'a': createnew();
                       break;
             case 'p': if(t2==0 && t3==0)
                          printnodes(1,cnt);
                       else
                          printnodes(t2,t3);
                       break;
             case 's': save();
                       break;
             case 'i': insertnodes(t2);
                       break;
             case 'd': if(t3==0)
                          deletenodes(t2,t2);
                       else
                          deletenodes(t2,t3);
                       break;
             case 'c': if(t4==0)
                          {
                           copynodes(t2,t2);
                           pastenodes(t3);
                          }
                       else
                          {
                           copynodes(t2,t3);
                           pastenodes(t4);
                          }
                       break;
             case 'm': if(t4==0)
                          movenodes(t2,t2,t3);
                       else
                          movenodes(t2,t3,t4);
                       break;
             case 'f': disppattern(tok2);
                       break;
             case 'q': save();
                       exit(0);
             default : printf("\nInvalid choice");
            }
     }
}
```


