```
/**
* NSG ACADEMY
* 2017
* LINE EDITOR
**/


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

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

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

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()
{
 node* temp;
 char str[80];
 printf("\nEnter some text(press stop)\n");
 gets(str);
 while(strcmp(str,"stop")!=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);
      }
}

node* findnode(int pos)
{
```

```
 int i=1;
 node* p;
 p=first;
 while(p!=NULL && i<pos)
      {
        p=p->next;
        i++;
      }
 return p;
}


void printnode(int m,int n)
{
 node* p;
 int i=0;
 p=findnode(m);
 while(p!=NULL && i<=n-m)
       {
        printf("\n%d %s",i+m,p->line);
        p=p->next;
        i++;
       }
}

void save()
{
 node* p;
 fp=fopen(fname,"w");
 p=first;
 while(p!=NULL)
      {
        strcat(p->line,"\n");
        fputs(p->line,fp);
        p=p->next;
      }
 fclose(fp);
}



void main(int argc,char* argv[])
{
 int n,x,y;
 char cmd[20],t1[20],t2[20],t3[20];

 strcpy(fname,argv[1]);

 fp=fopen(fname,"r");
 if(fp==NULL)
   {
    printf("\n%s file is not found\n",argv[1]);
    createnew();
    printf("\nNumber of lines is %d\n",cnt);
   }
 else
   {
    printf("\n%s file is found\n",argv[1]);
    create();
    printf("\nNumber of lines is %d\n",cnt);
    fclose(fp);
   }
 while(1)
       {
        printf("\n");
        printf("$");
        gets(cmd);
        n=sscanf(cmd,"%s %s %s",t1,t2,t3);
        switch(n)
              {
                case 1: if(strcmp(t1,"p")==0)
                           printnode(1,cnt);
```

```
                        else if(strcmp(t1,"a")==0)
                               createnew();
                             else if(strcmp(t1,"s")==0)
                                    save();
                                  else if(strcmp(t1,"q")==0)
                                         {
                                           printf("\nBYE_BYE");
                                           save();
                                           exit(0);
                                         }
                        break;
                  case 3: if(strcmp(t1,"p")==0)
                             {
                               x=atoi(t2);
                               y=atoi(t3);
                               printnode(x,y);
                             }
                        break;
                }

        }
}
```