#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX 20

typedef struct node{

  char data[MAX];

  struct node \*link;

}\*LIST;

void init(LIST \*L);

void insert(LIST \*L, char \*str);

void show(LIST L);

void del(LIST \*L, char \*str);

void edit(LIST \*L, char \*oldstr, char \*newstr);

void init(LIST \*L){

  \*L = NULL;

}

void insert(LIST \*L, char \*str){

  LIST \*trav;

  LIST temp;

  for(trav = L; (\*trav)!= NULL; trav = &(\*trav)->link);

  temp = malloc(sizeof(struct node));

  if (temp != NULL){

    strcpy(temp->data, str);

    temp->link = NULL;

    \*trav = temp;

  }else{

    printf("\nFAILED TO ALLOCATE MEMORY\n");

  }

}

void insertSorted(LIST \*L, char \*str){

  LIST \*trav;

  LIST temp;

  for(trav = L; (\*trav)!= NULL && (strcmp((\*trav)->data, str) < 0); trav = &(\*trav)->link);

  temp = malloc(sizeof(struct node));

  if (temp != NULL){

    strcpy(temp->data, str);

    temp->link = (\*trav);

    \*trav = temp;

  }else{

    printf("\nFAILED TO ALLOCATE MEMORY\n");

  }

}

void show(LIST L){

  LIST trav;

  for (trav = L; trav!=NULL; trav = trav->link){

    printf("%s \n", trav->data);

    if (trav->link != NULL){

      //printf("--> ");

    }

  }

}

void del(LIST \*L, char \*str){

  LIST \*trav;

  LIST temp;

  for(trav = L; (\*trav)!= NULL && (strcmp((\*trav)->data,str)!=0); trav = &(\*trav)->link){}

  if (\*trav!=NULL){

    temp = \*(trav);

    \*trav = (\*trav)->link;

    free(temp);

  }else{

    printf("\nNOT FOUND!\n");

  }

}

void edit(LIST \*L, char \*oldstr, char \*newstr){

  LIST \*trav;

  LIST temp;

  for(trav = L; (\*trav)!= NULL && (strcmp((\*trav)->data,oldstr)!=0); trav = &(\*trav)->link){}

  if (\*trav!=NULL){

    strcpy((\*trav)->data, newstr);

  }else{

    printf("\nNOT FOUND!\n");

  }

}

void main(){

  LIST \*head;

  init(&head);

  insertSorted(&head, "Farah");

  insertSorted(&head, "Sakura");

  insertSorted(&head, "Kyle");

  insertSorted(&head, "Christian");

  insertSorted(&head, "Nicholas");

  insertSorted(&head, "Athena");

  insertSorted(&head, "Rei");

  insertSorted(&head, "asdf");

  show(head);

  printf("\n");

  del(&head, "asdf");

  show(head);

  printf("\n");

  edit(&head, "Christian", "Christian Maderazo");

  show(head);

}