#include<stdio.h>

#include<stdlib.h>

struct listNode{

char data;

struct listNode\* nextPtr;

};

typedef struct listNode ListNode;

typedef ListNode \*ListNodePtr;

void insert(ListNodePtr \*sPtr, char value);

char delete(ListNodePtr \*sPtr, char value);

int isEmpty(ListNodePtr sPtr);

void printList(ListNodePtr currentPtr);

void instructions(void);

int main(void)

{

ListNodePtr startPtr = NULL;

int choice;

char item;

instructions();

printf("? ");

scanf("%d", &choice);

while (choice){

switch (choice){

case 1:

printf("Enter a character: ");

scanf("\n%c", &item);

insert(&startPtr, item);

printList(startPtr);

break;

case 2:

if (!isEmpty(startPtr)){

printf("Enter character to be deleted: ");

scanf("\n%c", &item);

if (delete(&startPtr, item)){

printf("%c deleted \n", item);

printList(startPtr);

}

else{

printf("%c not found.\n\n", item);

}

}

else{

printf("List is empty.\n\n");

}

break;

default:

printf("Invalid choice.\n\n");

instructions();

break;

}

printf("? ");

scanf("%d", &choice);

}

printf("End of run.\n");

return 0;

}

void instructions(void)

{

printf("Enter your choice:\n"

" 1 to insert an element into the list.\n"

" 2 to delete an element from the list.\n"

" 3 to end.\n");

}

void insert(ListNodePtr\* sPtr, char value)

{

ListNodePtr newPtr;

ListNodePtr previousPtr;

ListNodePtr currentPtr;

newPtr = malloc(sizeof(ListNode));

if (newPtr != NULL){

newPtr->data = value;

newPtr->nextPtr = NULL;

previousPtr = NULL;

currentPtr = \*sPtr;

while (currentPtr != NULL&&value > currentPtr->data){

previousPtr = currentPtr;

currentPtr = currentPtr->nextPtr;

}

if (previousPtr == NULL){

newPtr->nextPtr = \*sPtr;

\*sPtr = newPtr;

}

else{

previousPtr->nextPtr = newPtr;

newPtr->nextPtr = currentPtr;

}

}

else{

printf("%c not inserted. No memory available.\n", value);

}

}

char delete(ListNodePtr\* sPtr, char value)

{

ListNodePtr previousPtr;

ListNodePtr currentPtr;

ListNodePtr tempPtr;

if (value == (\*sPtr)->data){

tempPtr = \*sPtr;

\*sPtr = (\*sPtr)->nextPtr;

free(tempPtr);

return value;

}

else{

previousPtr = \*sPtr;

currentPtr = (\*sPtr)->nextPtr;

while (currentPtr != NULL&&currentPtr->data != value){

previousPtr = currentPtr;

currentPtr = currentPtr->nextPtr;

}

if (currentPtr != NULL){

tempPtr = currentPtr;

previousPtr->nextPtr = currentPtr->nextPtr;

free(tempPtr);

return value;

}

}

return '\0';

}

int isEmpty(ListNodePtr sPtr)

{

return sPtr == NULL;

}

void printList(ListNodePtr currentPtr)

{

if (currentPtr == NULL){

printf("List is empty.\n\n");

}

else{

printf("The list is:\n");

while (currentPtr != NULL){

printf("%c --> ", currentPtr->data);

currentPtr = currentPtr->nextPtr;

}

printf("NULL\n\n");

}

}