```
// 171805024 Nagihan BAZ
#include <stdio.h>
#include <stdlib.h>


struct flights {
    char name[30];
    int number;
    struct flights* nextPtr;
};

typedef struct flights flights;
typedef flights* flightsPtr;


void insert(flightsPtr* sPtr, char name[30]);
char delete(flightsPtr* sPtr, char name[30]);
int isEmpty(flightsPtr sPtr);
void printList(flightsPtr currentPtr);
void instructions(void);

int main(void)
{
    flightsPtr startPtr = NULL;
    unsigned int choice;
    char item;

    instructions();
    printf("%s", "Enter 1 to insert a flight. ");
    scanf("%u", &choice);
    printf("%s", "Enter 2 to delete a flight.");
    scanf("%u", &choice);
    printf("%s", "Enter 3 to sort the flights according to their numbers in ascending
order. ");
    scanf("%u", &choice);
    printf("%s", "Enter 4 to print the information of all flights.");
    scanf("%u", &choice);
    printf("%s", "Enter 5 to end.");
    scanf("%u", &choice);



    while (choice != 5) {

        switch (choice) {
        case 1:
            printf("%s", "Enter a flight: ");
            scanf("\n%c", &item);
            insert(&startPtr, item);
            printList(startPtr);
            break;
        case 2:

            if (!isEmpty(startPtr)) {
                printf("%s", "Enter flight to be deleted: ");
                scanf("\n%c", &item);
                if (delete(&startPtr, item)) {
                    printf("%c Flight is deleted.\n", item);
                    printList(startPtr);
                }
                else {
                    printf("%c Not found.\n\n", item);
                }
            }
            else {
                puts("List is empty.\n");
            }

            break;
        case 3:
            printf("%s", "Enter to sort the flights according to their numbers in
ascending order:");
            scanf("\n%c", &item);
            printList(startPtr);

            break;
        case 4:
            printf("%s", "Enter to print the information of all flights:");
            scanf("\n%c", &item);
            printList(startPtr);

            break;
        default:
            puts("Invalid choice.\n");
            instructions();
            break;
        }

        printf("%s", "Enter your choice.");
        scanf("%u", &choice);
    }

    puts("End of run.");
}

void instructions(void)
{
    puts("Enter your choice:\n"
        "   1 to insert a flight into the list.\n"
        "   2 to delete a flight from the list.\n"
        "   3 to sort the flights according to their numbers in ascending order.\n"
        "   4 to print the information of all flights.\n"
        "   5 to end.\n");
}

void insert(flightsPtr* sPtr, char name[30])
{
    flightsPtr newPtr;
    flightsPtr previousPtr;
    flightsPtr currentPtr;

    newPtr = malloc(sizeof(flights));

    if (newPtr != NULL) {
        newPtr->name[30] = name[30];
        newPtr->nextPtr = NULL;

        previousPtr = NULL;
        currentPtr = *sPtr;


        while (currentPtr != NULL && name[30] > currentPtr->name[30]) {
            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", name[30]);
    }
}

char delete(flightsPtr* sPtr, char name[30])
{
    flightsPtr previousPtr;
    flightsPtr currentPtr;
    flightsPtr tempPtr;


    if (name[20] == (*sPtr)->name[30]) {
        tempPtr = *sPtr;
        *sPtr = (*sPtr)->nextPtr;
        free(tempPtr);
        return name[30];
    }
    else {
        previousPtr = *sPtr;
        currentPtr = (*sPtr)->nextPtr;


        while (currentPtr != NULL && currentPtr->name[30] != name[30]) {
            previousPtr = currentPtr;
            currentPtr = currentPtr->nextPtr;
        }


        if (currentPtr != NULL) {
            tempPtr = currentPtr;
            previousPtr->nextPtr = currentPtr->nextPtr;
            free(tempPtr);
            return name[30];
        }
    }
    return '\0';
}

int isEmpty(flightsPtr sPtr)
{
    return sPtr == NULL;
}


void printList(flightsPtr currentPtr)
{

    if (isEmpty(currentPtr)) {
        puts("List is empty.\n");
    }
    else {
        puts("The list is:");


        while (currentPtr != NULL) {
            printf("%c --> ", currentPtr->name[30]);
            currentPtr = currentPtr->nextPtr;
        }

        puts("NULL\n");
    }
}
```