

**数据结构 课程设计报告**

**项 目 名 称： 学生宿舍管理系统\_\_\_\_\_\_\_ \_\_**

**专 业 班 级：2022级计算机科学与技术（商业智能系统开发）**

**小 组 成 员： 段昊男 \_**

**任 课 教 师： 毕丙申\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_**

**学 年 学 期： 2022-2023学年 第二学期\_\_\_\_\_ \_\_**

项目基本信息

|  |  |
| --- | --- |
| 项目名称 | 学生宿舍管理系统 |
| 项目简介 | 该项目设计主要是对学生宿舍的基本信息进行各种功能操作。初始化录入宿舍信息、删除、查询、浏览、添加、排序等操作来与用户进行交互并管理宿舍信息。 |
| 小组成员 | 段昊男 |
| 任务分工 | 段昊男：整个项目 |

项目评定成绩记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指导教师意见 | 系统完成情况：优 良 中 及格 不及格 | | | |
| 报告完成情况：优 良 中 及格 不及格 | | | |
| 答辩评定成绩 | 团队整体成绩： | | | |
| 成员成绩 | 2022105380129 | 段昊男 |  |
| 备 注 | 1.在指导教师意见相应成绩等级上划✓。  2.团队成绩按百分制。 | | | |

# 目录

目录

[1 目录 3](#_Toc121009097)

[2 任务要求介绍 4](#_Toc121009098)

[2.1 题目 4](#_Toc121009099)

[2.2 功能 4](#_Toc121009100)

[2.3 基本要求 5](#_Toc121009101)

[3 概述 5](#_Toc121009102)

[3.1 系统介绍 5](#_Toc121009103)

[3.2 Main函数运行界面 6](#_Toc121009104)

[3.3 Main函数功能流程图 7](#_Toc121009105)

[4 总体设计 8](#_Toc121009107)

[4.1 功能模块图 8](#_Toc121009108)

[4.2 主函数说明 8](#_Toc121009109)

[4.3 流程图 10](#_Toc121009110)

[5 详细设计 10](#_Toc121009111)

[5.1 void printList(LinkList L); 10](#_Toc121009112)

[5.2 void insertElem(LinkList L, LinkList Elem); 11](#_Toc121009113)

[5.3 LinkList creatIncreLink(); 12](#_Toc121009114)

[5.4 void deleteElem(LinkList L, int i); 12](#_Toc121009115)

[5.5 int searchNum(LinkList L, int n); 13](#_Toc121009116)

[5.6 int delNum(LinkList L, int n); 14](#_Toc121009117)

[5.7 int reviseRoom(LinkList L, int n); 14](#_Toc121009118)

[5.8 void addNum(LinkList L); 15](#_Toc121009119)

[5.9 void sortNum(LinkList L); 17](#_Toc121009120)

[6 运行结果与调试 18](#_Toc121009121)

[6.1 编译运行 18](#_Toc121009122)

[6.2 录入初始化宿舍信息 18](#_Toc121009123)

[6.3 删除宿舍信息 19](#_Toc121009124)

[6.4 查询宿舍信息 19](#_Toc121009125)

[6.5 修改宿舍信息 19](#_Toc121009126)

[6.6 浏览宿舍信息 20](#_Toc121009127)

[6.7 添加宿舍信息 20](#_Toc121009128)

[6.8 排序宿舍信息 20](#_Toc121009129)

[6.9 退出 21](#_Toc121009130)

[7 参考资料 21](#_Toc121009134)

# 任务要求介绍

## 题目

学生宿舍管理系统设计

## 功能

实现简单的学生宿舍基本信息管理，宿舍的基本信息包括楼号、房间号、面积、所容纳人数、已入住人数等。

## 基本要求

设计简单的菜单，能够进行系统功能选择

实现信息的录入功能:

在已有信息的基础上添加新的记录:

删除指定房间号的记录;

修改指定房间号的记录;

实现信息的浏览功能:

按房间号查询功能:

按已入住人数排序功能;

# 概述

## 系统介绍

学生宿舍管理系统对于一个学校来说是必不可少的组成部分。目前好多学校还停留在宿舍管理人员手工记录数据的最初阶段，手工记录对于规模小的学校来说还勉强可以接受，但对于学生信息量比较庞大，需要记录存档的数据比较多的高校来说，人工记录是相当麻烦的。而且当查找某条记录时，由于数据量庞大，还只能靠人工去一条条的查找，这样不但麻烦还浪费了许多时间，效率也比较低。当今社会是飞速进步的世界，原始的记录方式已经被社会所淘汰了，计算机化管理正是适应时代的产物。信息世界永远不会是一个平静的世界，当一种技术不能满足需求时，就会有新的技术诞生并取代旧技术。21世纪的今天,信息社会占着主流地位，计算机在各行各业中的运用已经得到普及，自动化、信息化的管理越来越广泛应用于各个领域。我们针对如此，设计了一套学生宿舍管理系统。学生宿舍管理系统采用的是计算机化管理，系统做的尽量人性化，使用者会感到操作非常方便，管理人员需要做的就是将数据输入到系统的数据库中去。由于数据库存储容量相当大，而且比较稳定，适合较长时间的保存，也不容易丢失。这无疑是为信息存储量比较大的学校提供了一个方便、快捷的操作方式。本系统具有运行速度快、安全性高、稳定性好的优点，并且具备完善的报表生成、修改功能，能够快速的查询学校所需的住宿信息。学生宿舍，是大学生在高校校园里一个重要的学习、生活、交往的空间环境，大学生，大约有2/3的时间是在宿舍环境里度过的。随着高校官学规模的扩大，在校学生的学历层次、思想状态、文化水平呈多样化、复杂化趋势，这在很大程度上给学生管理工作带来许多新的问题。随着学生人数的增多，高校学生宿舍管理的信息量日益庞大，而目前我校学生宿舍管理仍采用传统的工作方式——账本化的工作模式，这种模式获取信息慢，查阅困难，准确性差，很难适应高水平宿舍管理的需要，同时带来了大量的人力的浪费，对于宿管的工作带来很多不便。基于此，我们想到需要开发一个“学生宿舍管理系统”来提高宿舍的管理效率，改善管理方式，使其与现代信息化的趋势接轨。由于数据库的存储容量相当大，而且比较稳定，适合较长时间的保存，也不容易丢失。这无疑是为信息存储量比较大的学校提供了一个方便、快捷的操作方式。

## Main函数运行界面

在main函数中，主要提供视图选项以供用户进行选择操作，主要有如下选项：  
[1]录入并初始化宿舍信息  
[2]删除宿舍信息  
[3]查询宿舍信息  
[4]修改宿舍信息  
[5]浏览宿舍信息  
[6]添加宿舍信息  
[7]按入住人数排序  
[0]退出系统

## Main函数功能流程图

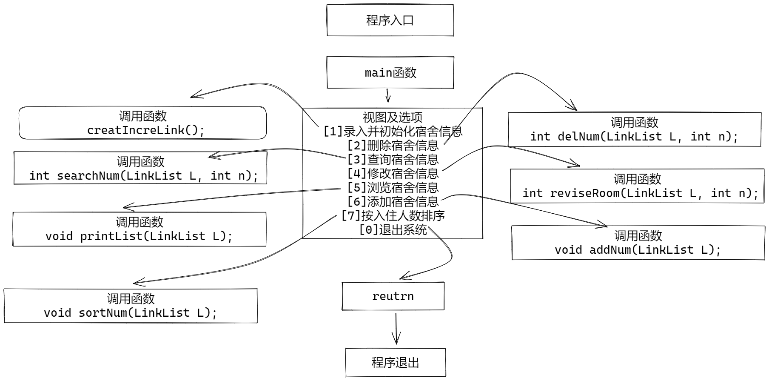


图 3‑1

各函数具体流程图见目录详细设计-函数流程图

# 总体设计

## 功能模块图

1录入宿舍信息

2删除宿舍信息

3查询宿舍信息

4修改宿舍信息

5浏览宿舍信息

学

生

宿

舍

管

理

系

统

6添加宿舍信息

7排序遍历宿舍信息

0退出系统

图 4‑1

各函数具体流程图见目录详细设计-函数流程图

## 主函数说明

主函数采用简单视图，用while循环来获取用户输入的选项，得以控制程序的进行。用户启动程序后将看到一个视图，显示了各个功能及编号。若用户需要操作，需要输入相应编号调用到对应的功能。



图 4‑2



图 4‑3



图 4‑4

## 流程图

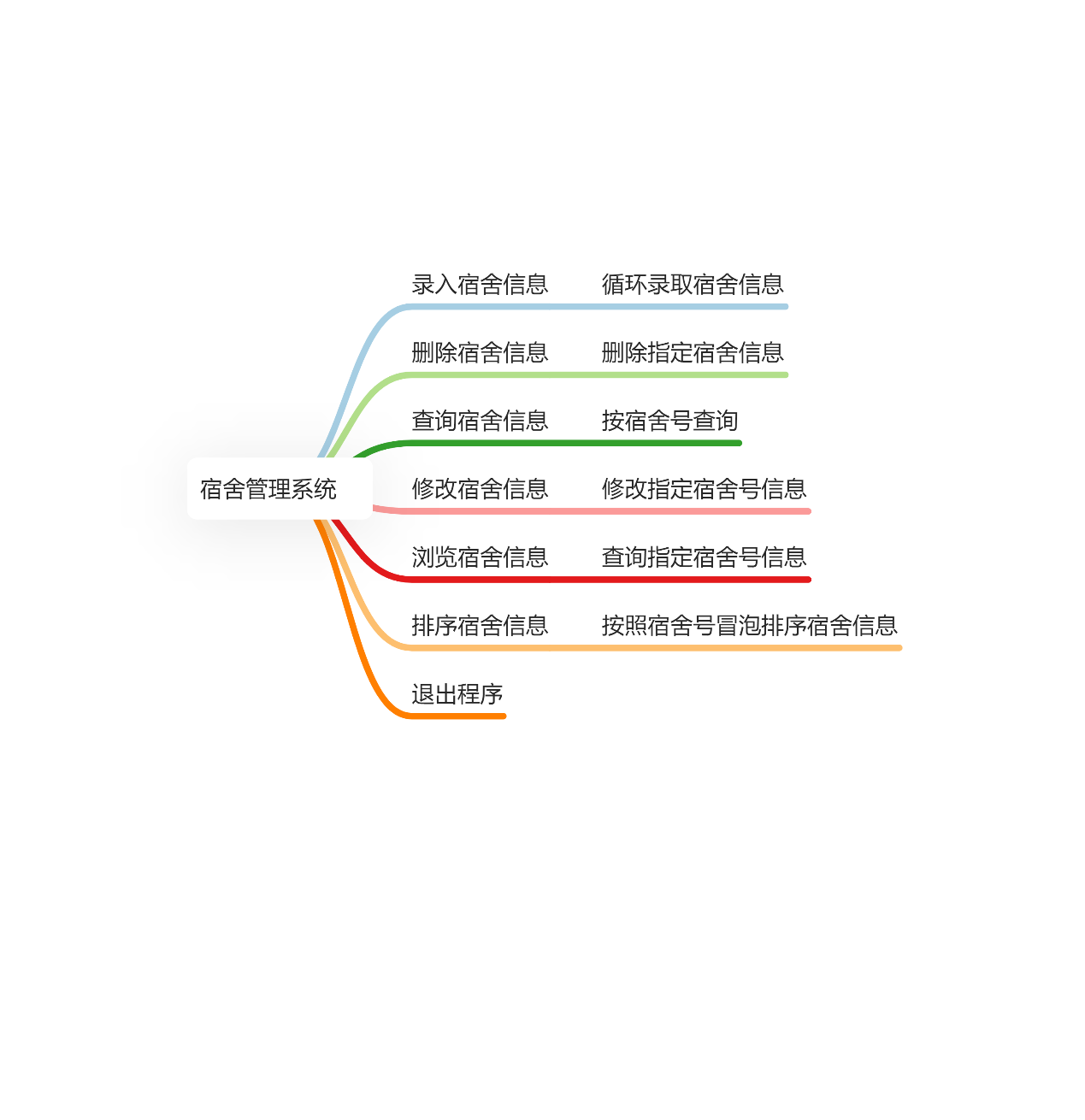


图 4‑5

# 详细设计

## void printList(LinkList L);

此函数为查看宿舍信息的函数。其功能主要为采用链表形式逐个查看每个节点的宿舍信息。该函数流程图如下：

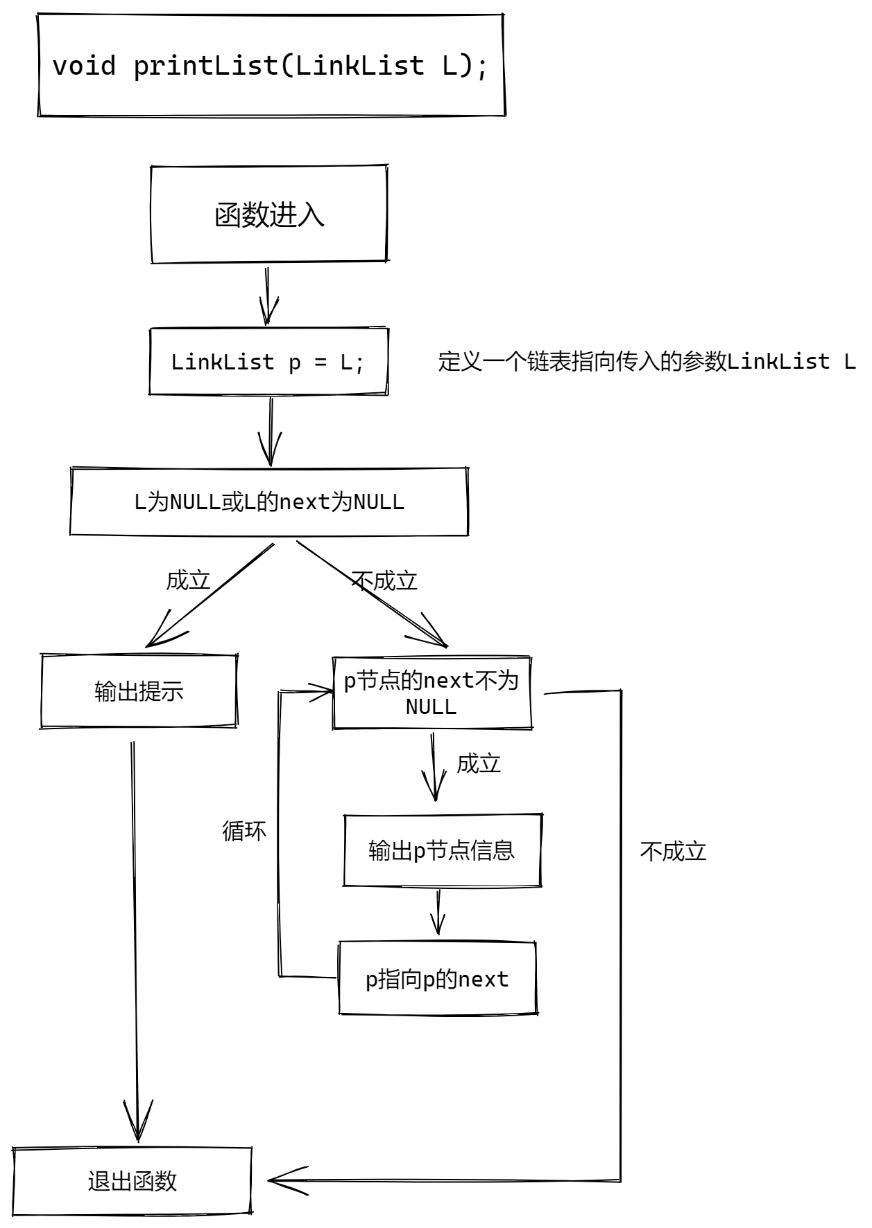


图 5‑1

## void insertElem(LinkList L, LinkList Elem);

该函数功能为向链表L中插入新的节点Elem，L是存储宿舍信息的链表，Elem是新的宿舍信息节点。该函数流程图如下：

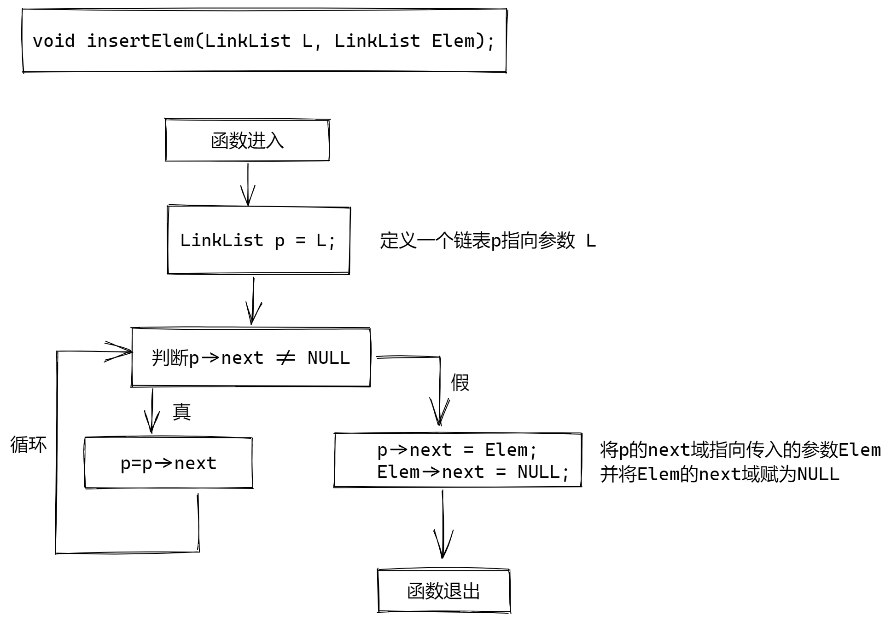


图 5‑2

## LinkList creatIncreLink();

此函数功能为创建两个链表，初始化空间及初始化宿舍信息，其中L用于存储宿舍信息，p作为节点存储信息，其后方便将p数据插入到L中。该函数流程图如下：

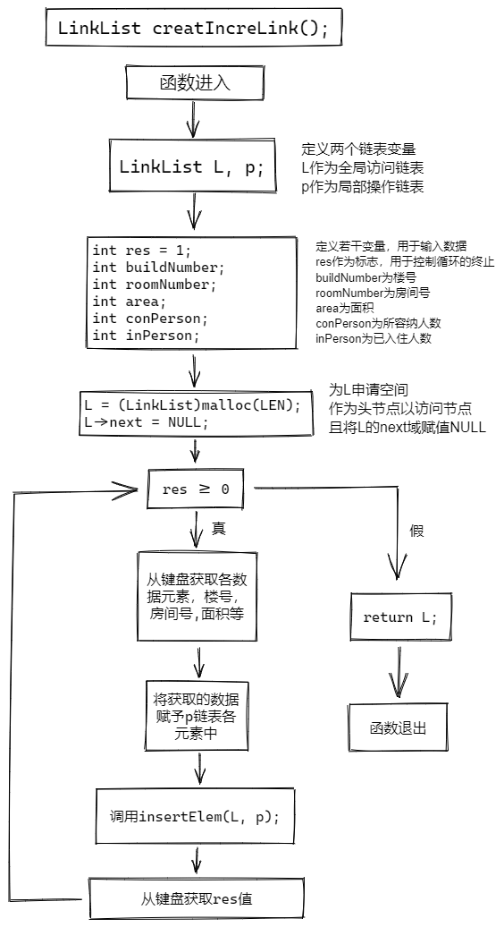


图 5‑3

## void deleteElem(LinkList L, int i);

此函数功能为从L表中删除编号为i的宿舍信息。传入两个参数，L为存储宿舍信息的链表，i为删除的宿舍编号。流程图如下：

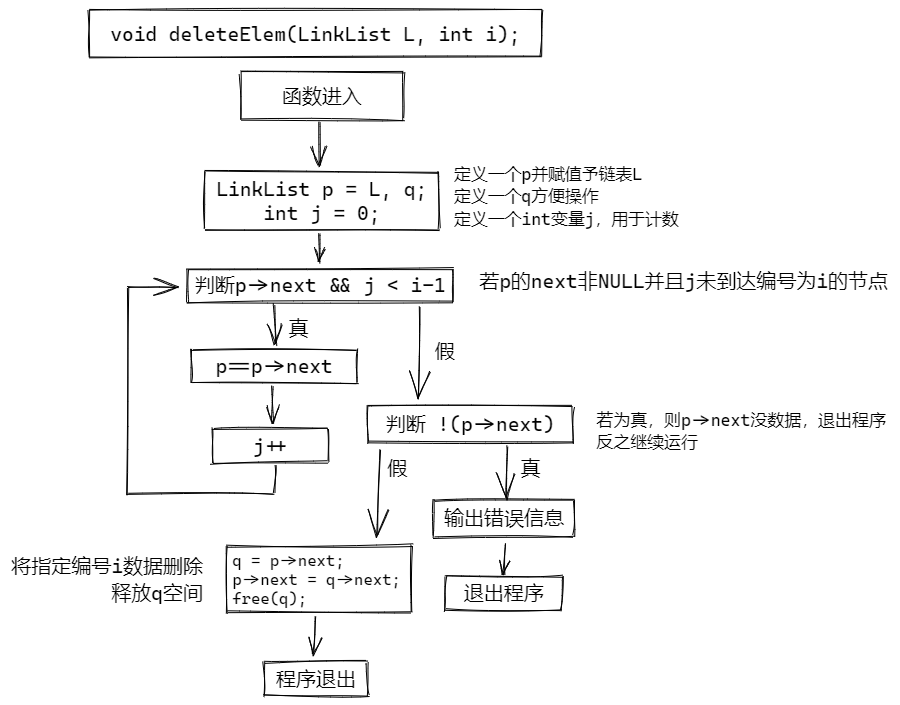


图 5‑4

## int searchNum(LinkList L, int n);

该函数功能为按照宿舍号查询宿舍信息。传入两个参数，L为存储宿舍信息的链表，n为宿舍号（对应roomNumber）。函数流程图如下：

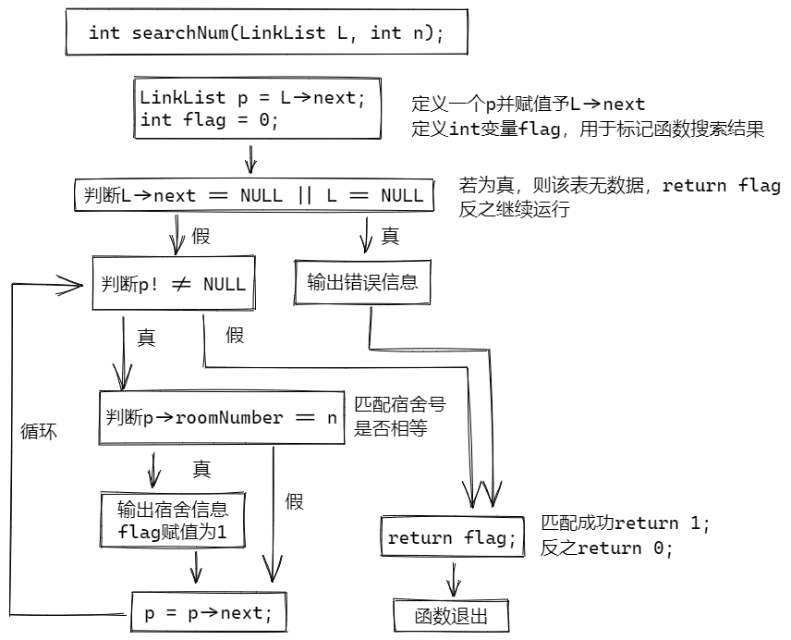


图 5‑5

## int delNum(LinkList L, int n);

该函数功能为按照房间号删除信息。传入两个参数，L为存储宿舍信息的链表，n为需要删除的宿舍号（对应roomNumber）。函数流程图如下：

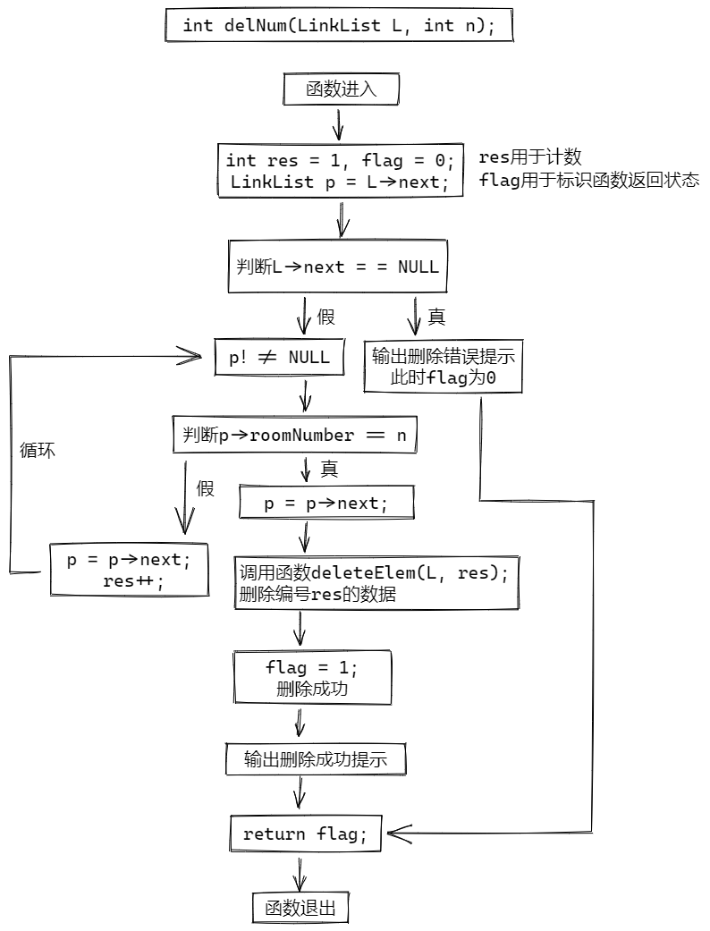


图 5‑6

## int reviseRoom(LinkList L, int n);

该函数功能为修改指定编号为n的元素数据。传入两个参数，L为存储宿舍信息的链表，n为需要修改的宿舍号（对应roomNumber）。函数流程图如下：

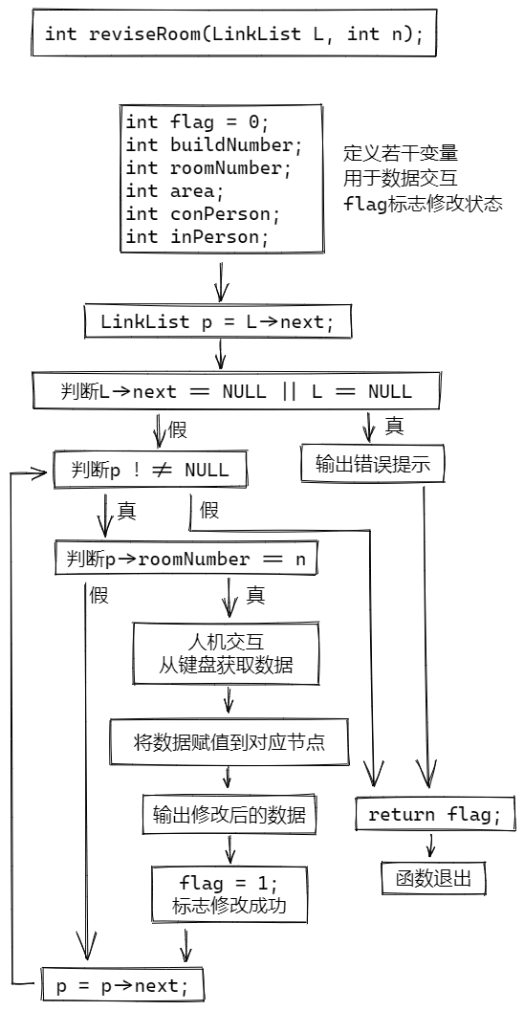


图 5‑7

## void addNum(LinkList L);

该函数功能为在宿舍已有信息基础上添加信息，传入一个参数，L为存储宿舍信息的链表，函数流程图如下：

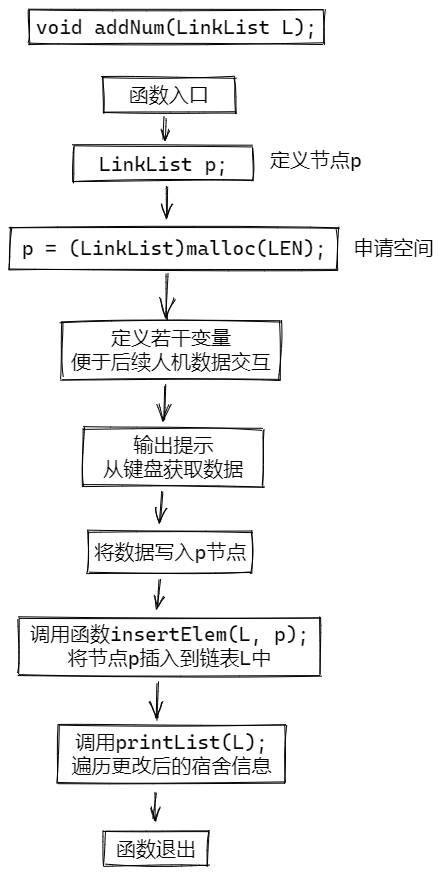


图 5‑8

## void sortNum(LinkList L);

此函数功能为按已入住人数排序，传入一个参数，L为存储宿舍信息的链表，函数流程图如下：

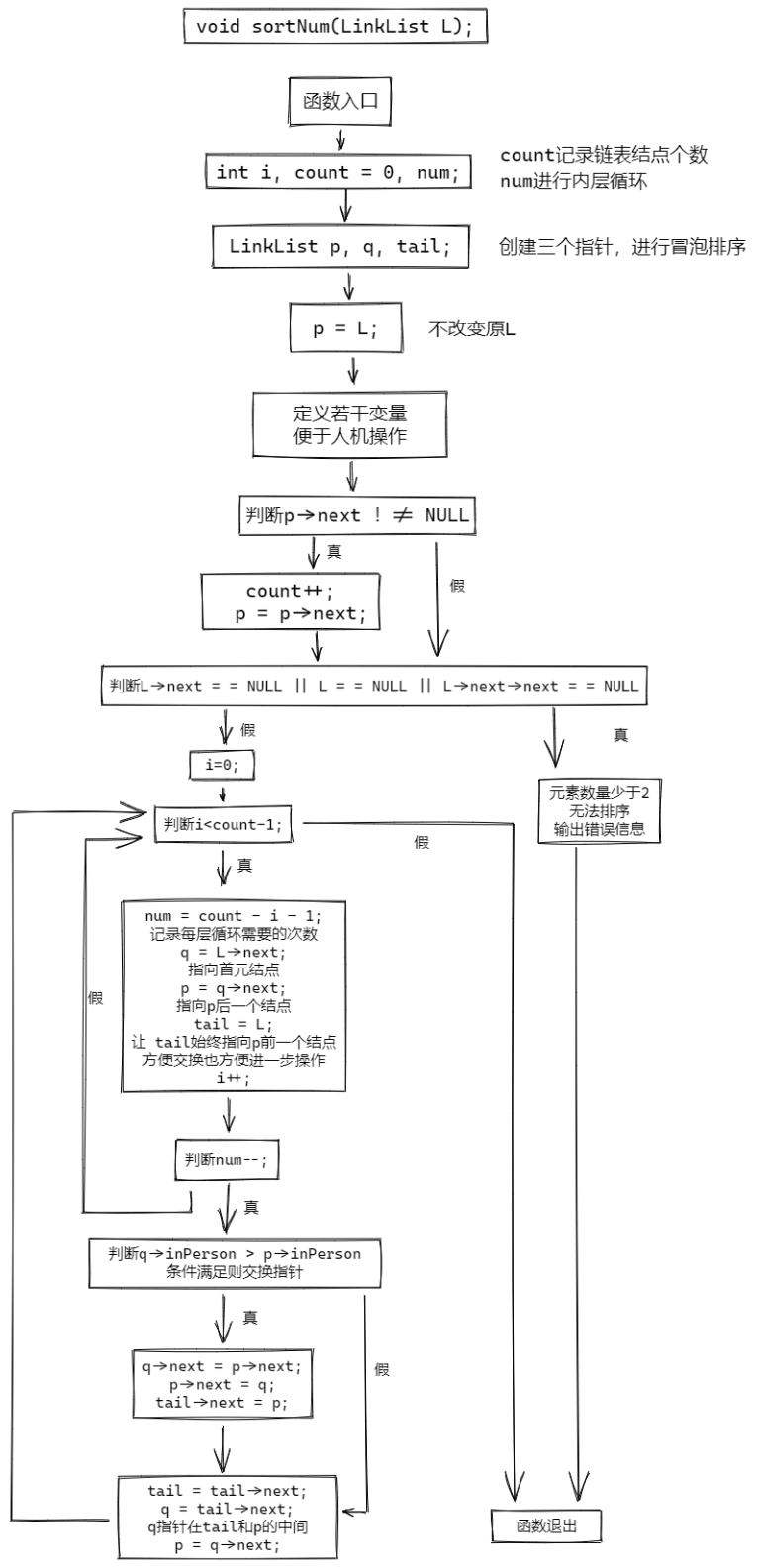


图 5‑9

# 运行结果与调试

## 编译运行

显示界面，与用户进行信息交互，初始化，删除，查询，修改，浏览，添加，按入住人数排序，退出系统。

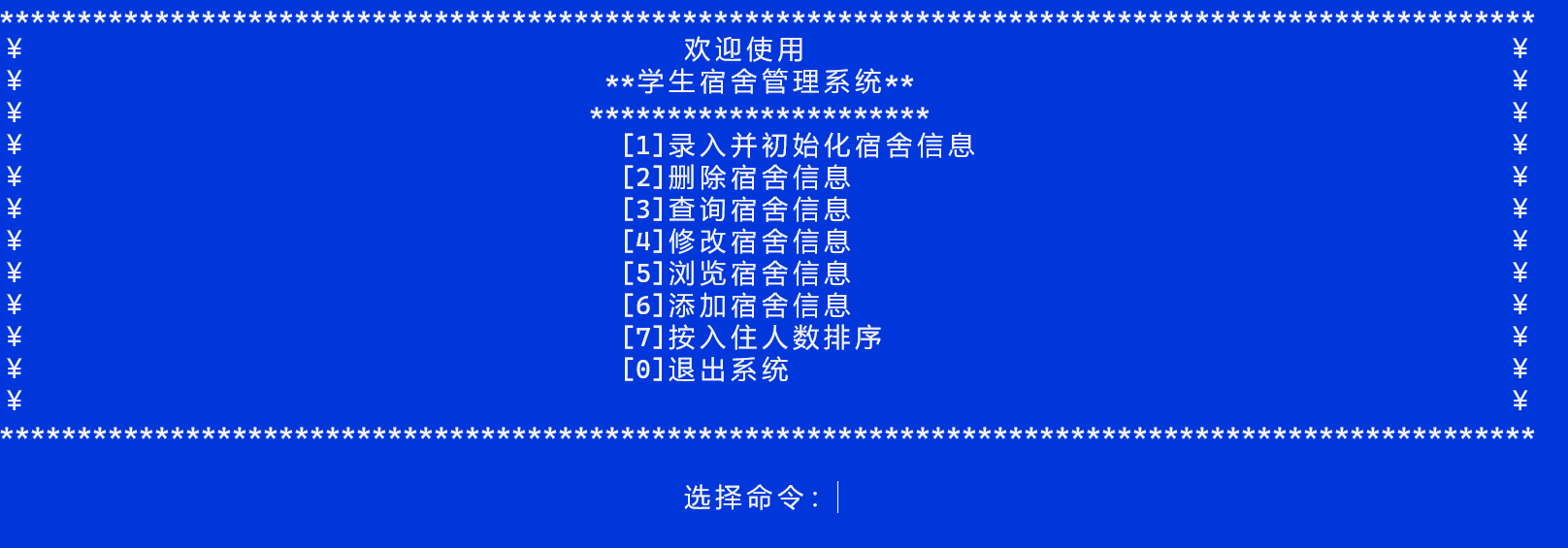


图 6‑1

## 录入初始化宿舍信息

选择命令1，初始化并录入信息，分别输入楼号，宿舍号，面积，所容纳人数，已入住人数，输入-1则退出录入信息。

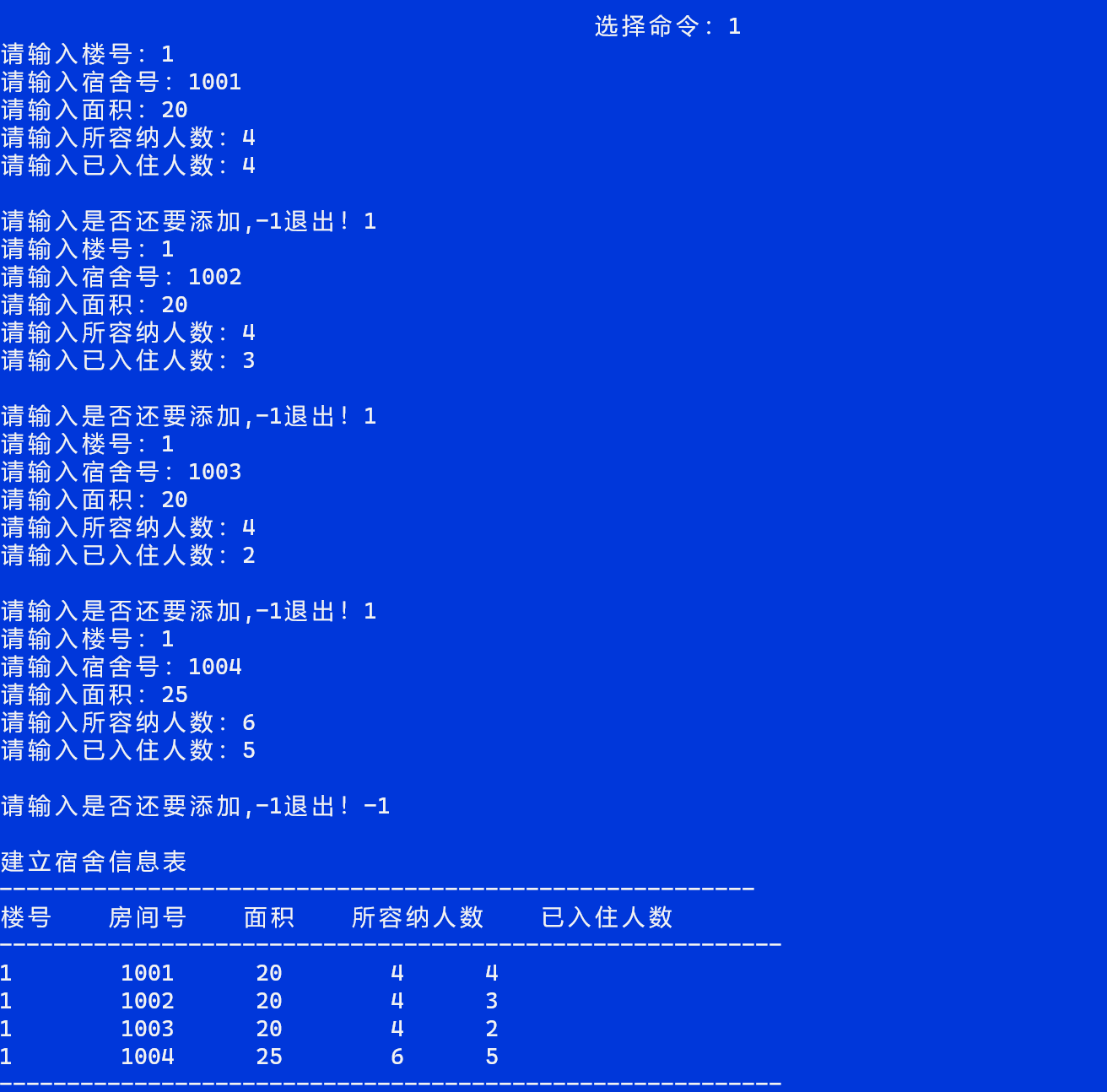


图 6‑2

## 删除宿舍信息

将宿舍已有信息进行删除操作，输入房间号，将此房间号中的所有信息删除。



图 6‑3

## 查询宿舍信息

将已有宿舍信息进行查询，输入房间号，将此房间号中的所有信息查询显示出来。



图 6‑4

## 修改宿舍信息

将要修改的宿舍的所有信息进行修改，按房间号进行查找要修改的信息，然后依次输入要修改的宿舍的楼号、宿舍号、面积、所容纳人数、已入住人数。

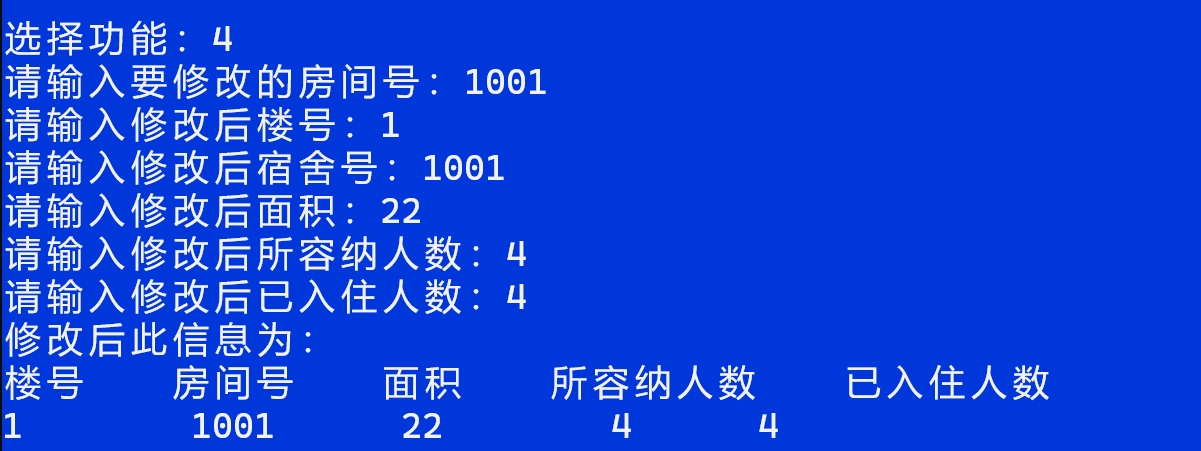


图 6‑5

## 浏览宿舍信息

浏览宿舍的所有信息，将整个宿舍的所有信息都显示出来。

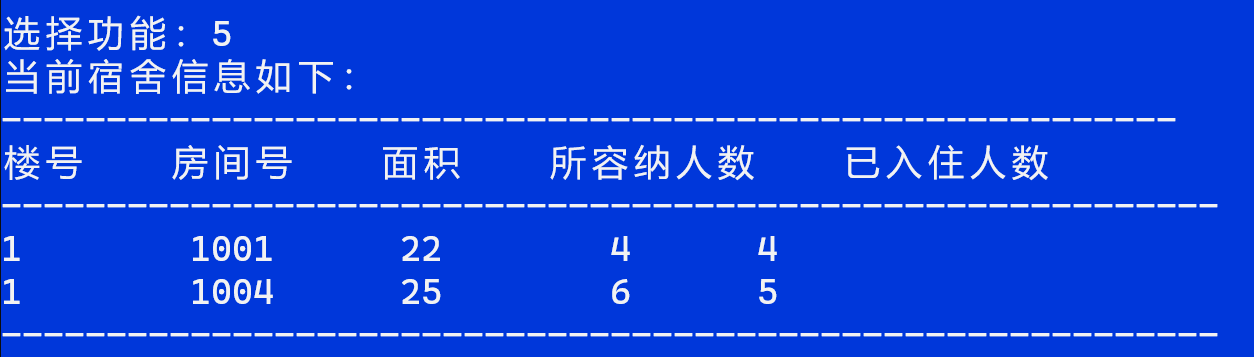


图 6‑6

## 添加宿舍信息

在原有宿舍信息中添加新的信息，新的信息跟原宿舍信息拥有相同的功能及信息，都包括宿舍的楼号、宿舍号、面积、所容纳人数、已入住人数。

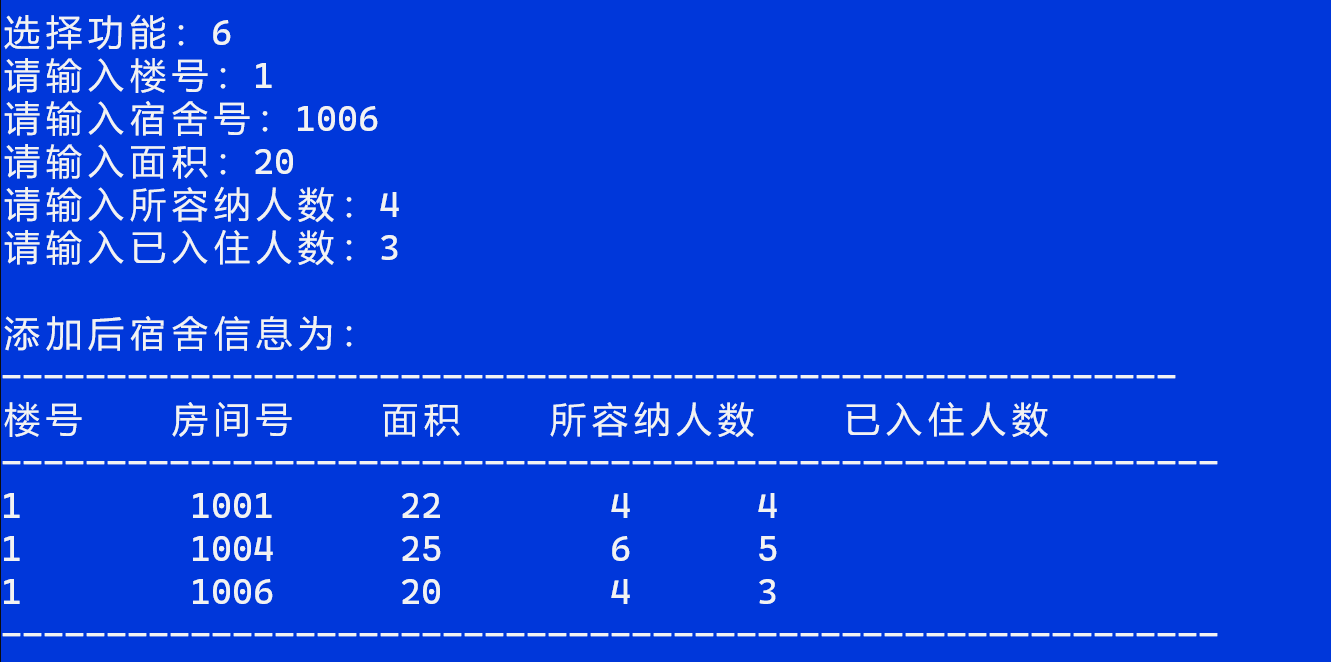


图 6‑7

## 排序宿舍信息

将宿舍中所有信息进行排序，按照已入住人数从小到大进行排序。



图 6‑8

## 退出

# 参考资料

1. 严蔚敏 / 吴伟民.《数据结构(c语言版)》.北京:人民邮电出版社，2022.23~67
2. 严蔚敏.《清华大学计算机系列教材：数据结构(C语言版)》.北京:人民邮电出版社， 2021.44~98
3. 谭浩强. 《c程序设计(第五版) (谭浩强编著) 》.北京：清华大学出版社，2021.101~133
4. Thomas H.Cormen / Charles E.Leiserson / Ronald L.Rivest / Clifford Stein. 《算法导论 原书第三版 》.( 殷建平 / 徐云 / 王刚 / 刘晓光 / 苏明 / 邹恒明 / 王宏志).北京：机械工业出版社，2012.78~34
5. [美] Brian W. Kernighan / [美] Dennis M. Ritchie. 《C程序设计语言（第2版·新版） 》.北京：机械工业出版社，2016.56~144
6. 查询信息的网站：csdn、github、菜鸟教程、博客园、百度等。

**源代码:**

/\*

功能:实现简单的学生宿舍基本信息管理，

宿舍的基本信息包括楼号、房间号、面积、所容纳人数、已入住人数等。

\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define LEN sizeof(LNode) //结点结构定义

typedef struct LNode{ //宿舍系统结点

int buildNumber; //楼号

int roomNumber; //房间号

int area; //面积

int conPerson; //所容纳人数

int inPerson; //已入住人数

struct LNode \*next; //指针域

} LNode, \*LinkList;

// 菜单

void menu(){

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("￥ 欢迎使用 ￥\n");

printf("￥ \*\*学生宿舍管理系统\*\* ￥\n");

printf("￥ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ￥\n");

printf("￥ [1]录入并初始化宿舍信息 ￥\n");

printf("￥ [2]删除宿舍信息 ￥\n");

printf("￥ [3]查询宿舍信息 ￥\n");

printf("￥ [4]修改宿舍信息 ￥\n");

printf("￥ [5]浏览宿舍信息 ￥\n");

printf("￥ [6]添加宿舍信息 ￥\n");

printf("￥ [7]按入住人数排序 ￥\n");

printf("￥ [0]退出系统 ￥\n");

printf("￥ ￥\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf(" 选择命令：");

}

//打印头结点地址为L的链表，查看宿舍信息

void printList(LinkList L){

LinkList p = L; //p指向L，即头结点

printf("\n--------------------------------------------------------\n");

printf("楼号 房间号 面积 所容纳人数 已入住人数\n");

printf("----------------------------------------------------------\n");

if(L == NULL || L->next == NULL){

printf("该宿舍管理系统中没有信息！！！\n");

} else{

while(p->next != NULL){

printf("%d %d %d %d %d\n", p->next->buildNumber, p->next->roomNumber, p->next->area, p->next->conPerson, p->next->inPerson);

p = p->next;

}

}

printf("----------------------------------------------------------\n");

return;

}

//添加一个元素（指针）

void insertElem(LinkList L, LinkList Elem){

LinkList p = L;

while(p->next != NULL){ //p->next是指向了第一个结点(首元结点)

p = p->next;

}

p->next = Elem;

Elem->next = NULL;

printf("\n");

}

//创建一个存放信息表，返回头结点地址,输入-1结束

LinkList creatIncreLink(){

LinkList L, p;

// LinkList s = L;

int res = 1; //res让输入表示循环是否终止

int buildNumber; //楼号

int roomNumber; //房间号

int area; //面积

int conPerson; //所容纳人数

int inPerson; //已入住人数

L = (LinkList)malloc(LEN); //生成新的结点作为头结点，用头指针指向头结点

L->next = NULL; //置空头结点的指针域，即首元结点为NULL

while(res >= 0){

printf("请输入楼号：");

scanf\_s("%d", &buildNumber);

printf("请输入宿舍号：");

scanf\_s("%d", &roomNumber);

printf("请输入面积：");

scanf\_s("%d", &area);

printf("请输入所容纳人数：");

scanf\_s("%d", &conPerson);

printf("请输入已入住人数：");

scanf\_s("%d", &inPerson);

p = (LinkList)malloc(LEN); //创建新的结点

p->buildNumber = buildNumber;

p->roomNumber = roomNumber;

p->area = area;

p->conPerson = conPerson;

p->inPerson = inPerson;

insertElem(L, p);

printf("请输入是否还要添加,-1退出！");

scanf\_s("%d", &res);

}

return (L);

}

// 从系统中删除第i个元素

void deleteElem(LinkList L, int i){

LinkList p = L, q;

int j = 0;

//先查找出来这个元素

while(p->next && j < i-1){

p = p->next;

j++;

}

//判断i是否合法，i既不能大于元素的个数，也不能小于等于0

if(!(p->next)){

printf("第%d个元素删除失败\n", i); //因为后面没结点的值了

return;

}

q = p->next;

p->next = q->next;

free(q); //释放删除的结点

}

// 按房间号查询

int searchNum(LinkList L, int n){

//先查找出来然后再显示

LinkList p = L->next;

int flag = 0;

if(L->next == NULL || L == NULL){

printf("该宿舍系统中没有元素，查找失败\n");

} else{

while(p != NULL){

if(p->roomNumber == n){

printf("楼号 房间号 面积 所容纳人数 已入住人数\n");

printf("%d %d %d %d %d\n", p->buildNumber, p->roomNumber, p->area, p->conPerson, p->inPerson);

flag = 1;

}

p = p->next;

}

}

return flag;

}

// 按房间号删除

int delNum(LinkList L, int n){

int res = 1, flag = 0;

LinkList p = L->next; //指针p指向首元结点

if(L->next == NULL){

printf("该宿舍系统中没有元素，删除失败\n");

}else{

while(p != NULL){

if(p->roomNumber == n){

p = p->next;

deleteElem(L, res);

flag = 1;

}else{

p = p->next;

res++;

}

}

printf("删除成功！\n");

}

return flag;

}

//按房间号修改元素

int reviseRoom(LinkList L, int n){

int flag = 0;

int buildNumber; //楼号

int roomNumber; //房间号

int area; //面积

int conPerson; //所容纳人数

int inPerson; //已入住人数

LinkList p = L->next;

if(L->next == NULL || L == NULL){

printf("该宿舍系统中没有元素，查找失败\n");

}else{ //先找到要修改的房间号

while(p != NULL){

if(p->roomNumber == n){ //查找成功开始修改

printf("请输入修改后楼号：");

scanf\_s("%d", &buildNumber);

printf("请输入修改后宿舍号：");

scanf\_s("%d", &roomNumber);

printf("请输入修改后面积：");

scanf\_s("%d", &area);

printf("请输入修改后所容纳人数：");

scanf\_s("%d", &conPerson);

printf("请输入修改后已入住人数：");

scanf\_s("%d", &inPerson);

p->buildNumber = buildNumber;

p->roomNumber = roomNumber;

p->area = area;

p->conPerson = conPerson;

p->inPerson = inPerson;

printf("修改后此信息为：\n");

printf("楼号 房间号 面积 所容纳人数 已入住人数\n");

printf("%d %d %d %d %d\n", p->buildNumber,

p->roomNumber, p->area, p->conPerson, p->inPerson);

flag = 1;

}

p = p->next;

}

}

return flag;

}

//已有信息基础上添加信息

void addNum(LinkList L){

LinkList p;

p = (LinkList)malloc(LEN);

int buildNumber; //楼号

int roomNumber; //房间号

int area; //面积

int conPerson; //所容纳人数

int inPerson; //已入住人数

//输入新结点的信息

printf("请输入楼号：");

scanf\_s("%d", &buildNumber);

printf("请输入宿舍号：");

scanf\_s("%d", &roomNumber);

printf("请输入面积：");

scanf\_s("%d", &area);

printf("请输入所容纳人数：");

scanf\_s("%d", &conPerson);

printf("请输入已入住人数：");

scanf\_s("%d", &inPerson);

p->buildNumber = buildNumber;

p->roomNumber = roomNumber;

p->area = area;

p->conPerson = conPerson;

p->inPerson = inPerson;

insertElem(L, p); // 插入信息到L链表末尾

printf("添加后宿舍信息为：");

printList(L);

}

//按已入住人数排序

void sortNum(LinkList L){

int i, count = 0, num; //count记录链表结点个数,num进行内层循环

LinkList p, q, tail; //创建三个指针，进行冒泡排序

p = L;

int buildNumber; //楼号

int roomNumber; //房间号

int area; //面积

int conPerson; //所容纳人数

int inPerson; //已入住人数

//计算结点个数

while(p->next != NULL){

count++;

p = p->next;

}

if(L->next == NULL || L == NULL || L->next->next == NULL){

printf("该宿舍系统中元素小于2，无法排序\n");

}else{

for(i=0; i<count-1; i++){

num = count - i - 1; //记录每层循环需要的次数

q = L->next; //指向首元结点

p = q->next; //指向p后一个结点

tail = L; //让 tail始终指向p前一个结点，方便交换也方便进一步操作

while(num--){ //内层循环

if(q->inPerson > p->inPerson){ //条件满足则交换指针

q->next = p->next;

p->next = q;

tail->next = p;

}

tail = tail->next;

q = tail->next; //q指针在tail和p的中间

p = q->next;

}

}

}

}

//主函数

int main(){

LinkList L = NULL, p;

int menuNum; //menuNum指的是菜单选项

char temp;

system("color 1f");

menu(); //菜单显示

scanf\_s("%d",&menuNum);

//开始执行

while(menuNum){

switch(menuNum){

case 1:{

L = creatIncreLink();

printf("\n建立宿舍信息表");

printList(L);

break;

}

case 2:{

int n, f;

printf("请输入要删除的房间号：");

scanf\_s("%d", &n);

f = delNum(L, n);

if(f != 0){

printf("删除后：\n");

printList(L);

}else{

printf("无该宿舍号，删除失败！\n");

}

break;

}

case 3:{

int room, s;

printf("请输入要查找的房间号：");

scanf\_s("%d", &room);

s = searchNum(L, room);

if(s == 0){

printf("没有此房间，查找失败！\n");

}

break;

}

case 4:{

int room, s;

printf("请输入要修改的房间号：");

scanf\_s("%d", &room);

s = reviseRoom(L, room);

if(s == 0){

printf("没有此房间，修改失败！\n");

}

break;

}

case 5:{

printf("当前宿舍信息如下：");

printList(L);

break;

}

case 6:{

addNum(L);

break;

}

case 7:{

printf("排序后宿舍信息如下：");

sortNum(L);

printList(L);

break;

}

case 0:{

exit (0);

}

default:{

printf("\n没有此功能，重新输入！");

}

}

printf("\n选择功能：");

scanf\_s("%d",&menuNum);

}

return 0;

}