**北京邮电大学软件学院**

**2016-2017学年第1学期实验报告**

**课程名称：\_算法与数据结构\_**

**实验名称：\_线性表\_**

**实验完成人： \_刘浩博\_**

**姓名：\_刘浩博\_ 学号：\_2015212086\_ 成绩：\_\_\_\_\_\_\_\_\_\_\_**

**指导老师： \_贾红娓\_**

**日期： 2016 年 10 月 27 日**

⼀.**实验⽬的**

本次实验的主要⽬的在于熟悉线性表的基本运算在两种存储结构上的实现，其中以熟

悉各种链表的操作为侧重点。同时，通过本次实验帮助学⽣复习⾼级语⾔的使⽤⽅法。

**⼆. 实验内容**

**1.**城市链表

**[问题描述]**

将若⼲城市的信息，存⼊⼀个带头结点的单链表。结点中的城市信息包括：城市名，

城

市的位置坐标。要求能够利⽤城市名和位置坐标进⾏有关查找、插⼊、删除、更新等操作。

**[基本要求]**

　　（1） 给定⼀个城市名，返回其位置坐标；

　　（2） 给定⼀个位置坐标P和⼀个距离D，返回所有与P的距离⼩于等于D的城市。

**[测试数据]**

由学⽣依据软件⼯程的测试技术⾃⼰确定。注意测试边界数据。

**2.** 约瑟夫环

**[问题描述]**

约瑟夫（Joeph）问题的⼀种描述是：编号为1,2,…,n的n个⼈按顺时针⽅向围坐⼀圈，

每⼈持有⼀个密码（正整数）。⼀开始任选⼀个正整数作为报数上限值m，从第⼀个⼈开

!2

始按顺时针⽅向⾃1开始顺序报数，报到m时停⽌报数。报m的⼈出列，将他的密码作为新

的m值，从他在顺时针⽅向上的下⼀个⼈开始重新从1报数，如此下去，直⾄所有⼈全部出

列为⽌。试设计⼀个程序求出出列顺序。

**[基本要求]**

利⽤单向循环链表存储结构模拟此过程，按照出列的顺序印出各⼈的编号。

**[测试数据]**

　　m的初值为20；密码：3，1，7，2，4，8，4（正确的结果应为6，1，4，7，2，3，

5）。

**[实现提⽰]**

程序运⾏后⾸先要求⽤户指定初始报数上限值，然后读取各⼈的密码。设n≤30。

**三. 实验环境**

VC6.0

**四. 实验结果**

两个程序均能正确实现实验目标，并且无错误情况出现

**五. 附录**

**City.c:**

#define N 100

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

#include <iostream>

using namespace std;

typedef struct Data

{

char city;

float x;

float y;

}data;

typedef struct CNode

{

data data;

struct CNode \*next;

}node, \*City;

void CreateList(City &L)//创建链表

{

int value;

printf("开始创建，初始要输入几个城市：\n");

scanf("%d", &value);

L = (City)malloc(sizeof(node));

L->next = NULL;

for (int i = 0; i<value; i++)

{

City p = (City)malloc(sizeof(node));

printf("请输入%d个城市的数据：\n", i+1);

cin >> p->data.city >> p->data.x >> p->data.y;

p->next = L->next;

L->next = p;

}

}

void search(City L)//查找

{

char name;

City p;

p = L;

printf("请输入要查找的城市：\n");

scanf("%c", &name);

printf("坐标为：");

while (p)

{

if (p->data.city == name)

printf("%f,%f", p->data.x, p->data.y);

p = p->next;

}

}

void Insert(City L)//插入

{

City p = L->next;

City ss = (City)malloc(sizeof(node));

printf("请输入要插入的城市名及坐标：");

scanf("%c %f %f", &ss->data.city, &ss->data.x, &ss->data.y);

ss->next = p->next;

p->next = ss;

}

void Delete(City L)//删除

{

City q, p = L->next;

char c;

printf("请输入要删除的城市名：");

scanf("%c", &c);

while (p->data.city != c)

{

q = p;

p = p->next;

}

q->next = p->next;

free(p);

printf("已删除\n");

}

void print(City L)//打印

{

int i = 0;

City a;

char name[N];

float x[N], y[N];

a = L->next;

while (a)

{

name[i] = a->data.city;

x[i] = a->data.x;

y[i] = a->data.y;

a = a->next;

printf("%c %f,%f\n", name[i], x[i], y[i]);

i++;

}

}

void distance(City L)//距离

{

float s, d;

float x1, x2, y1, y2;

City p;

p = L->next;

printf("请输入位置坐标P:");

scanf("%f %f", &x1, &y1);

printf("请输入距离D:");

scanf("%f", &d);

while (p)

{

x2 = p->data.x;

y2 = p->data.y;

s = sqrt((x1 - x2)\*(x1 - x2) + (y1 - y2)\*(y1 - y2));

if (s <= d)

printf("%c %f,%f\n", p->data.city, x2, y2);

p = p->next;

}

}

void alter(City L)//修改

{

char a, b;

float c, d;

int i = 0;

City p = L;

printf("请输入需要替换的城市名：");

scanf("%c", &a);

while (p->data.city != a)

{

p = p->next;

}

printf("请输入新的城市名：");

scanf("%c", &b);

printf("请输入城市坐标：");

scanf("%f %f", &c, &d);

p->data.city = b; p->data.x = c; p->data.y = d;

}

int main()

{

City new\_l;

int i;

CreateList(new\_l);

do

{

printf("\n");

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

printf(" 请选择需要操作的功能： \n");

printf(" 1->输出城市链表信息\n");

printf(" 2->查找\n");

printf(" 3->插入\n");

printf(" 4->删除\n");

printf(" 5->修改\n");

printf(" 6->与坐标P距离小于D的城市\n");

printf(" 0->退出程序\n");

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

printf("\n");

printf("请输入您选择的功能号：");

scanf("%d", &i);

getchar();

if (i>0)

switch (i)

{

case 1:print(new\_l); break;

case 2:search(new\_l); break;

case 3:Insert(new\_l); break;

case 4:Delete(new\_l); break;

case 5:alter(new\_l); break;

case 6:distance(new\_l); break;

}

printf("\n");

if (i<0)

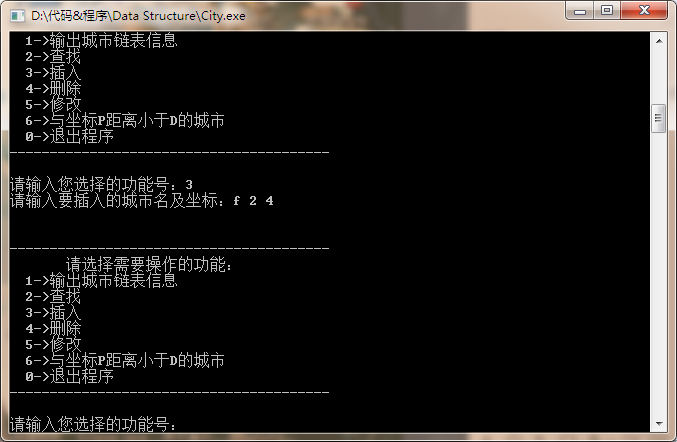
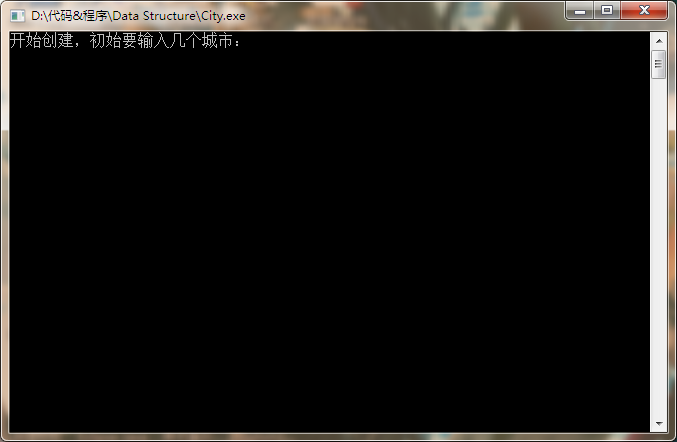
printf("数值不正确，请重新输入！\n");

} while (i != 0);

return 0;

}

截图：





**Joseph.cpp:**

#include<stdio.h>

#include<stdlib.h>

typedef struct LNode{

int data;

int number;

struct LNode \*next;

}LNode, \*LinkList;

void getResult(LinkList a, int x)

{

int i,j;

LinkList pri;

for (i = x; i>1; i--)

{

a = a->next;

}

pri = a;

a = a->next;

j = a->data;

printf("%d ", a->number);

pri->next = a->next;

free(a);

if (pri->next != pri)

getResult(pri,j);

else

{

printf("%d ", pri->number);

}

}

int main()

{

int n, m, i;

LinkList Joseph, top, P;

Joseph = (LinkList)malloc(sizeof(LNode));

top = (LinkList)malloc(sizeof(LNode));

printf("请输入总人数：");

scanf("%d", &n);

printf("请输入报数上限值：");

scanf("%d", &m);

printf("请输入第1个人所持的密码：");

P = (LinkList)malloc(sizeof(LNode));

scanf("%d", &P->data);

P->number = 1;

Joseph->next = P;

top = Joseph;

Joseph = Joseph->next;

for (i = 1; i<n; i++)

{

printf("请输入第%d个人所持的密码：", i + 1);

P = (LinkList)malloc(sizeof(LNode));

scanf("%d", &P->data);

P->number = i + 1;

Joseph->next = P;

Joseph = P;

}

Joseph->next = top->next;

getResult(top, m);

scanf("%d", &m);

return 0;

}

截圖：

