南京信息工程大学 数据结构 实验(实习)报告

实验(实习)名称 线性表的链式存储和操作 实验(实习)日期 10月25 得分 指导老师 马瑞

系 **计软院** 专业  **计算机科学与技术** 班级 5 姓名 刘祥宇学号 202183290006

一、实验目的

针对某个数值序列，设计一个哈希表，完成相应的建表和查表顺序。哈希函数用除留余数法

构造，用线性探测再散列的方法处理哈希地址冲突。

1. 实验内容与步骤

#include<iostream>

#include<malloc.h>

#define HashLength 13

#define NULLKEY 0

using namespace std;

typedef struct{

int key;

int compareTimes;

int flag;

}Elem;

typedef struct{

Elem \*base;

int count;

int length;

}HashList;

int Hash(int data){

return data%HashLength;

}

void InitHash(HashList &H) {

int i;

H.base= (Elem \*)malloc(HashLength\*sizeof(Elem));

H.length=HashLength;

for (i=0;i<HashLength; i++) {

H.base[i].key=NULLKEY;

}

}

void Insert(HashList &H,int data) {

int hashAddress=Hash(data);

while(H.base[hashAddress].key!=NULLKEY) {

H.base[hashAddress].flag=-1;

hashAddress=(++hashAddress)%HashLength;

}

H.base[hashAddress].key=data;

}

void CreateHashList(HashList &H){

int i,key;

int a[9]={23,5,17,12,26,31,13,4,6};

cout<<"哈希表长度为：9"<<endl;

for(i=0;i<9;i++){

key=a[i];

Insert(H,key);

}

}

void PrintHashList(HashList H){

cout<<"哈希表中元素为："<<endl;

int i;

for(i=0;i<H.length;i++){

if(H.base[i].key!=0){

cout<<H.base[i].key<<' ';

}

}

cout<<endl;

}

int SearchHashList(HashList H,int key){

int hashAddress=Hash(key);

int number=1;

while(H.base[hashAddress].key!=key){

number++;

hashAddress=(++hashAddress)%HashLength;

if(H.base[hashAddress].key==NULLKEY || hashAddress==Hash(key)){

return -1;

}

}

return hashAddress;

}

int SearchNumber(HashList H,int key){

int hashAddress=Hash(key);

int number=1;

while(H.base[hashAddress].key!=key){

number++;

hashAddress=(++hashAddress)%HashLength;

if(H.base[hashAddress].key==NULLKEY || hashAddress==Hash(key)){

return -1;

}

}

return number;

}

void Information(){

cout<<"============================================"<<endl;

cout<<" 1.创建哈希表"<<endl;

cout<<" 2.显示创建的哈希表内容"<<endl;

cout<<" 3.查找元素在哈希表中的位置（下标位置+1）"<<endl;

cout<<" 4.退出"<<endl;

cout<<"============================================"<<endl;

}

int main(){

int n,key,choose;

HashList H;

InitHash(H);

Information();

cout<<"请根据选项选择您的操作：\n";

cin>>choose;

while(choose!=4){

switch(choose){

case 1:{

CreateHashList(H);

break;

}

case 2:{

PrintHashList(H);

break;

}

case 3:{

cout<<"输入要查找的元素值:";

cin>>key;

n=SearchHashList(H,key);

if(n==-1){

cout<<"查询不到";

}else{

cout<<key<<"在哈希表中的下标位置是："<<n+1;

cout<<endl;

}

break;

}

default:{

cout<<"选择不正确，请重新输入！\n";

break;

}

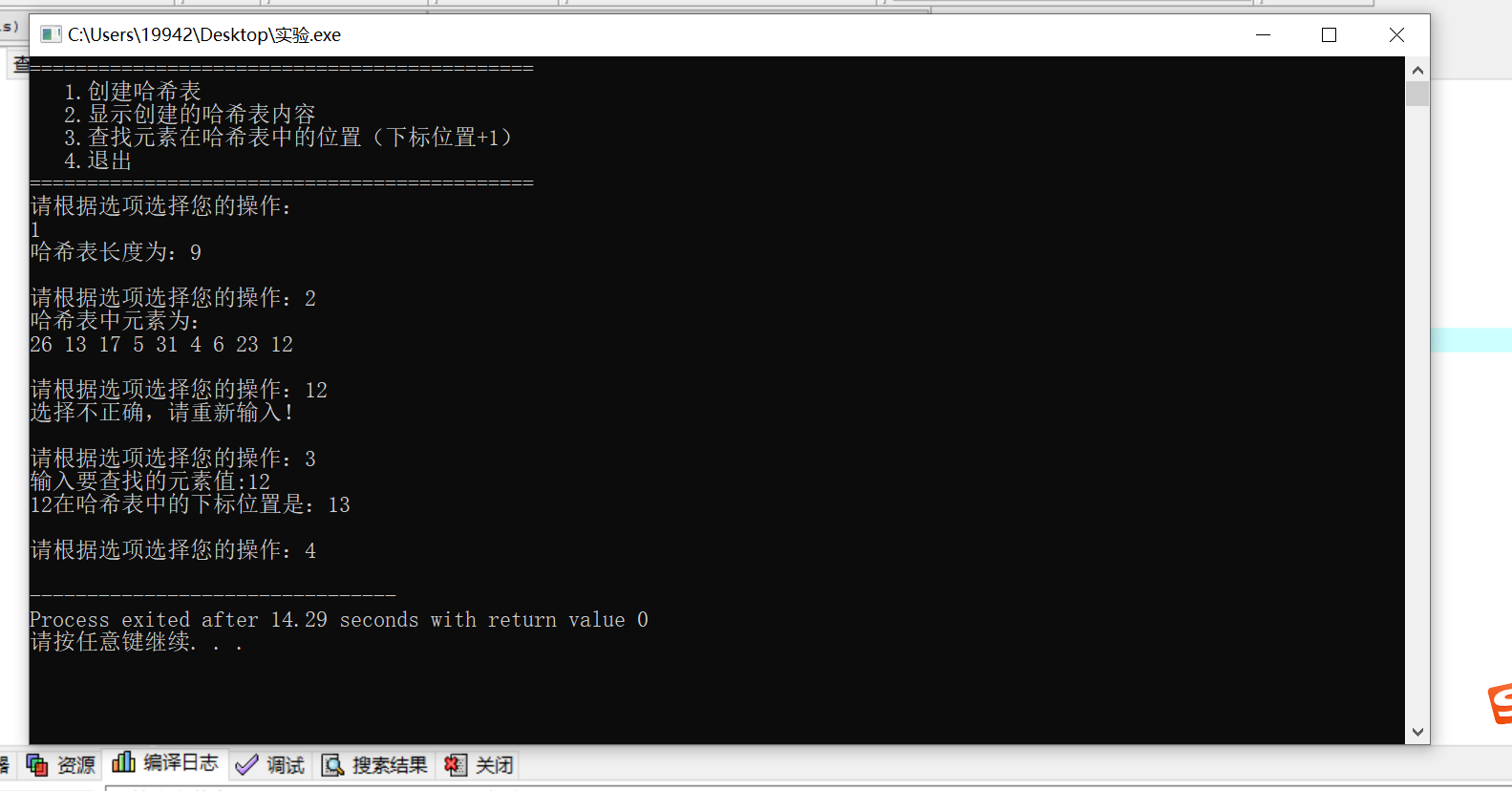
}

cout<<"\n请根据选项选择您的操作：";

cin>>choose;

}

}



**三、体会和总结**

1、哈希表底层数据结构是数组。

2、哈希表不能存储基本数据类型，哈希表存储的元素具有唯一性。

3、哈希表存储元素的过程是先判断要存储的元素与已有元素的hashcode是不是相同，如果不相同则会直接存储，如果相同就会继续调用equals方法判断，如果equals方法返回的是true则不能存储否则可以存储。