**哈希表的查找**

**一、实验目的**

1.掌握哈希表、哈希函数与哈希冲突的概念。

2.掌握哈希表的构造方法及其计算机的表示与实现。

3.掌握哈希表查找和插入算法的实现。

二、实验内容

以开放地址法中的线性探测再散列法处理冲突,实现哈希表的建立、査找和插入操作。

(1)设哈希表长为10,用除留余数法构造一个哈希函数H(key)=key%7

(2)输入哈希表中记录的个数n(n<=10)和各记录的关键字,然后以开放地址法的线性探测再散列法作为解决冲突的方法,建立一个哈希表,并输出已经建立的哈希表

(3)输入一个待查找记录的关键字key,完成开放地址哈希表的查找操作,如果查找成功,则函数返回查找到的记录在哈希表中的位置,否则给出查找失败的提示信息。

#include<stdio.h>

#define HASHSIZE 10//哈希表长

#define M 7 //小于哈希表长的最大质数

#define KEYNULL 0 //假设未存储关键字的位置的默认值

//哈希表数据结构

typedef struct{

int r[HASHSIZE];

int length;

}HashTable;

//初始化空的哈希表H，给其成员赋值

void InitHashTable(HashTable &H)

{

}

//除留余数法构造哈希函数

int HashFunc(int key)

{

return key%M;

}

/\*在哈希表H中查找关键字值为key的记录,

若探测到key对应的第一个空的地址空间,则用PosNULL返回位置,

若查找成功,函数返回待查找的记录在哈希表中的下标,

否则函数返回-1。\*/

int HashSearch(HashTable H,int key,int &PosNULL)

{

}

//查找不成功时,插入关键字值为key的记录到哈希表H中,并返回1,否则返回-1

int HashInsert(HashTable &H,int key)

{

}

//打印哈希表中所有的记录

void PrintHashTable(HashTable H)

{

}

void main()

{

HashTable H;

int key;

InitHashTable(H);

printf("请输入记录数：");

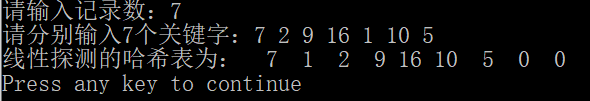
scanf("%d",&H.length);

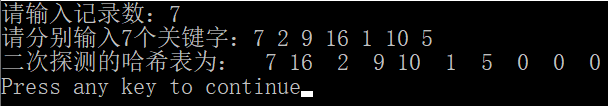
printf("请分别输入%d个关键字：",H.length);

//补充完整

PrintHashTable(H);

}





|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **地址** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **线性** | **7** | **1** | **2** | **9** | **16** | **10** | **5** | **0** | **0** | **0** |
| **二次** | **7** | **16** | **2** | **9** | **10** | **1** | **5** | **0** | **0** | **0** |