已知：

1、 数据：45、56、90、19、51、28、12、62、32、85

2、 散列表长m=14，散列函数为H(k) = key % 13

3、 使用线性探测来解决冲突，探测增量di序列为：｛1，2，3，4，…｝

在散列表中查找关键字k的算法思路：

1. 根据散列函数计算k的散列地址j;
2. 若b[j]为空，则查找失败；
3. 若b [j]等于k，则查找成功，返回关键字为k的记录在散列表中的下标j；
4. 若b[j]为不为空也不等于k，则按解决冲突的方法，让j指向下一个可能位置，转2；

将关键字k插入的算法思路：

1. 根据散列函数计算k的散列地址j;

2. 若b[j]为空，则将关键字k插入到b[j]；若b[j]不为空，转3

3. 按解决冲突的方法，让j指向下一个可插入位置，转2；

部分函数代码参考：

#define m 14

void ClearHashList(int \*b) //初始化散列表，初始所有数组元素值为0

{

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

b[i]=0;

}

void Insert(int \*b,int k) //插入k到散列表的散列位置

{

int j;

j=k%13;

int d=1;

while (b[j]!=0)

{

j=k%13;

j=(j+d)%m;

d++;

}

b[j]=k;

}

int Search(int \*b,int k) //查找k在散列表的位置

{

int j,d=1;

j=k%13;

while (b[j]!=0)

{

if (b[j]==k) return j;

else { j=k%13;j=(j+d)%m;d++;}

}

return -1;

}

**if(i%2==0) d=-(i/2)\*(i/2)**

**else d=(i+1)/2\*(i+1)/2**