Linear probing

```
#include<stdio.h>
 struct hash
{
int key,hkey;

} ht[10];

void insert(struct hash ht[],int size)
{
int ele,i;

printf("\nEnter the element to insert:");
scanf("%d",&ele);

i=0;
while(i<size)
{
if(ht[ele%size].key==-1)
{
ht[(ele%size+i)%size].key=ele;
ht[ele%size].hkey= ele%size;
}
else
{        i++;
ht[(ele%size+i)%size].key=ele;
```

```
ht[ele%size].hkey=ele%size;
}
i++;
}
}
 void search(struct hash ht[],int size)
{
int s,i;
printf("\nEnter the element to search:");
scanf("%d",&s);
if(ht[s%size].key==s)
printf("\nSearch element is available in %d location",s%size);
else
printf("\nSearch element is not available");

}
 void display(struct hash ht[],int size)
{
int i;
struct hash *p;
printf("\nhkey\t\tkey\n");
for(i=0;i<size;i++)
{
printf("%d\t\t%d\n",ht[i].hkey,ht[i].key);

}
```

```
}



main()
{
int size,i,ch;
printf("enter the size of the array:");
scanf("%d",&size);



for(i=0;i<size;i++)
ht[i].key = -1;
do
{
printf("\n1.Insert\n2.search\n3.display\n4.exit");
printf("\nEnter your choice:");
scanf("%d",&ch);
switch(ch)
{
case 1:insert(ht,size);
break;
case 2:search(ht,size);
break;
case 3:display(ht,size);
```

```
break;
case 4:exit(0);
default: printf("enter the valid choice");

}
}while(ch>=0&&ch<=4);
}
```

Output: