```
#include<stdio.h>
#define size 6
 struct hash
{
int key;
struct hash *next;
}*ht[size],*newnode,*p;
 void initially()
{
int i;
for(i=0;i<size;i++)
{
ht[i]=NULL;
}

}
 void insert()
{
int hi,ele;
printf("Enter the value of Key");
scanf("%d",&ele);

hi=ele%size;

newnode=(struct hash*)malloc(sizeof(struct hash));
newnode->key = ele;
newnode->next=NULL;

if(ht[hi]==NULL)
{
ht[hi]=newnode;
```

```
}
else
{
for(p=ht[hi];p->next !=NULL ;p=p->next);
p->next = newnode;
}
}
 void find()
{
int i,n,flag=0;
printf("enter element want to find");
scanf("%d",&n);
for(p=ht[n%size];p!=NULL;p=p->next)
{
if(p->key==n)
  { flag=1;
break;
}
}

if(flag==1)
printf("found");
else
printf("not found");

}
 void display()
{
int i;

printf("\n**The Hash table**\n\n HashTable[Hash Key]\t Chain of Key's\n");
```

```
for(i=0;i<size;i++)
{
if(ht[i] != NULL)
{
printf("Hash_Table[%d] = ",i);

for(p=ht[i]; p != NULL; p =p->next)
printf("->%d ",p->key);
}
printf("\n\n");
}
}
 main()
{
int cho,i,ele;

initially();

do
{
printf("\n** Main Menu **");
printf("\n1.Insert\n2.Find\n3.Display\n4.Exit\n");
printf("Enter your choice :");
scanf("%d",&cho);
switch(cho)
{
case 1:
insert();
break;
   case 2:
```

```
find();
break;
case 3:
display();
break;
case 4:
exit(0);
}
}while(cho!=4);
}
```

Output: