```
def is_full(dict,s):
    flag=0
    for i in range(s):
        if(dict[i]==0):
            flag+=1
    if(flag==s):
        return 1
    else:
        return 0



def linear(dict,s,n):
    test=is_full(dict,s)
    if(test==0 or s>=n):
        for i in range(n):
            id=int(input("Enter id:"))
            name=input("Enter name:")
            mod=id%s
            if(dict[mod]==0):
                dict[mod]=id,name
            else:
                while(dict[mod]!=0):
                    mod=(mod+1)%s
                dict[mod]=id,name
    else:
```

```
        print("Hash table is full or incapable to fill the data!!!")


def retrive(dict,s,id):
    a=id%s
    flag=-1
    b=0
    if(dict[a]!=0):
        while(b<s):
            if(dict[a][0]==id):
                print("Found")
                return
            else:
                flag=0
                b+=1
                a=(a+1)%s
        if(flag==0):
            print("Not")
    else:
        print("Not present")


def quadratic(dict,s,n):
    test=is_full(dict,s)
    if(test==0 or s>n):
        for i in range(n):
            id=int(input("Enter id:"))
```

```
            name=input("Enter name:")
            mod=id%s
            if(dict[mod]==0):
                dict[mod]=id,name
            else:
                a=1
                while(dict[mod]!=0):
                    mod=(mod+a**2)%s
                    a+=1
                dict[mod]=id,name
    else:
        print("Hash table is full or incapable to fill the data!!!")


def retrive_quad(dict,s,id):
    a=id%s
    flag=-1
    i=1
    b=(a-1)%s
    if(dict[a]!=0):
        while(dict[a]!=0 or a!=b):
            if(dict[a][0]==id):
                print("Found")
                return
            else:
                flag=0
```

```
            a=(a+(i**2))%s
            i+=1
        if(flag==0):
            print("Not")
    else:
        print("Not present")


def display(dict):
    for key in dict:
        print(key," ",dict[key])
```