

```

#include <iostream>
#include<bits/stdc++.h>
using namespace std;

template <typename K,typename V>
class HashNode{
public:
    K key;
    V value;
    HashNode(K key,V value){
        this->key=key;
        this->value=value;
    }
};

template <typename K,typename V>
class HashMap{
    HashNode<K,V> **arr;
    int capacity,size;
    HashNode<K,V> *dummy;
public:
    HashMap(){
        arr=new HashNode<K,V>* [capacity];
        for(size_t i=0;i<capacity;i++) arr[i]=nullptr;
        capacity=20;
        size=0;
        dummy=new HashNode<K,V>(-1,-1);
    }
    int HashCode(K key) {return key%capacity;}
    void InsertNode(K key,V value){
        if(size>=capacity) return;
        int HashIndex=HashCode(key);
        int count=0;
        while(arr[HashIndex]!=nullptr && arr[HashIndex]->key!=key &&
arr[HashIndex]->key!=-1){\
            if(++count>capacity) return;
            HashIndex++;
            HashIndex%=capacity;
        }
        if(arr[HashIndex]==nullptr || arr[HashIndex]->key==-1){
            HashNode<K,V>* newnode=new HashNode<K,V>(key,value);
            arr[HashIndex]=newnode;
            size++;
        }
    }
    V deleteNode(K key){
        int HashIndex=HashCode(key);
        int count=0;
        while(arr[HashIndex]!= nullptr && arr[HashIndex]->key!=key){
            if(++count>capacity) return 0;
            HashIndex++;
            HashIndex%=capacity;
        }
        if(arr[HashIndex]== nullptr) return 0;
        HashNode<K,V>* tmpnode=arr[HashIndex];

```

```

        arr[HashIndex]=dummy;
        size--;
        return tmpnode->value;
    }
    v findNode(K key){
        int HashIndex=HashCode(key);
        int count=0;
        while(arr[HashIndex]!= nullptr && arr[HashIndex]->key!=key){
            if(++count>capacity) return 0;
            HashIndex++;
            HashIndex%=capacity;
        }
        if(arr[HashIndex]== nullptr) return 0;
        return arr[HashIndex]->value;
    }
    int sizeofMap()
    {
        return size;
    }
    bool isEmpty()
    {
        return size == 0;
    }
    void display()
    {
        for(int i=0 ; i<capacity ; i++)
        {
            if(arr[i] != nullptr && arr[i]->key != -1)
                cout <<i<< " key = " << arr[i]->key
                    <<" value = "<< arr[i]->value << endl;
        }
    }
};

```