

```

#include <iostream>
#include <string>
using namespace std;

class Hashtable
{
    struct Entry
    {
        int key;
        string value;
        Entry* next;
        Entry(int id,string value):key(id),value(value),next(nullptr)
        {}
    };
    Entry **table;
    int size;
public:
    Hashtable(int size=10):size(size)
    {
        table=new Entry*[size];
        for (int i = 0; i < size; i++)
        {
            table[i]=nullptr;
        }
    }

    int hash(int key)
    {
        int offset = key % size;
        return offset;
    }

    void put (int key, const string &value)
    {
        int offset = hash(key);

        for (Entry *current=table[offset]; current; current=current->next)
        {
            if (current->key == key)
            {
                current->value = value;
                return;
            }
        }

        Entry *newEntry = new Entry(key, value);
        newEntry->next = table[offset];
        table[offset] = newEntry;
    }

    bool get (int key, string &result)
    {
        int offset = hash(key);

        for (Entry *current = table[offset]; current; current=current->next)
        {
            if(current->key==key)
            {
                result = current->value;
                return true;
            }
        }
        return false;
    }
}

```

```

void printDebug ()
{
    Entry *current=table[0];

    for(int i=0;i<size; i++)
    {
        cout<<"["<<i<<"] : ";
        for(Entry *current=table[i];current;current=current->next)
        {
            cout<<"["<<current->key<<" "<<current->value<<" "<<"] ";
        }

        cout<<endl;
    }
}

bool remove(int key)
{
    Entry *cur=table[hash(key)];
    Entry *back=table[hash(key)];

    for (;cur;cur=cur->next)
    {
        if (cur->key == key)
        {
            if (cur== table[hash(key)] )
            {
                table[hash(key)]=cur->next;
                delete cur;
            }
            else
            {
                back->next=cur->next;
                delete cur;
            }
            return 1;
        }
        back=cur;
    }
    return 0;
}

};

int main()
{
    Hashtable h;
    int key=1;
    string value="om";
    for(int i=0; key!=0 ;i++ )
    {
        cout<<"enter key and value : ";
        cin>>key;
        cin>>value;
        h.put(key,value);
    }

    h.printDebug();
    key=1;
    for(int i=0; key ;i++ )
    {
        cout<<"enter key to findout value : ";
        cin>>key;
    }
}

```

```

        h.get(key,value);
        cout<<value<<endl;
        value=" ";
    }

    key=1;
    for(int i=0; key!=0 ;i++ )
    {
        cout<<"enter key to delete value : ";
        cin>>key;
        h.remove(key);
        h.printDebug();
    }

}

```

OUTPUT

```

enter key and value : 1saf
enter key and value : 2asfasf
enter key and value : 2asfasdf
enter key and value : 8afdfaa
enter key and value : 0
0
[0] : [0 0 ]
[1] : [1 saf ]
[2] : [2 asfasdf ]
[3] :
[4] :
[5] :
[6] :
[7] :
[8] : [8 afdfaa ]
[9] :
enter key to findout value : 8
afdfaa
enter key to findout value : 0
0
enter key to delete value : 2
[0] : [0 0 ]
[1] : [1 saf ]
[2] :
[3] :
[4] :
[5] :
[6] :
[7] :
[8] : [8 afdfaa ]
[9] :
enter key to delete value :

```