```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;


namespace HashTable
{
    class HashTableItem<TKey, TValue>
    {
        public TKey Key { get; set; }
        public TValue Value { get; set; }

        public HashTableItem(TKey key, TValue value)
        {
            Key = key;
            Value = value;
        }
    }

    class HashTable<TKey, TValue>
    {
        private int size =0;
        private LinkedList<HashTableItem<TKey, TValue>>[] array;
        public HashTable(int size)
        {
          this.size = size;
            array = new LinkedList<HashTableItem<TKey, TValue>>[size];
        }

        private int Hash(TKey key)
        {
            return Math.Abs(key.GetHashCode() % size);
        }

        public int Add(TKey key, TValue value)
        {

            int index = Hash(key);

            if (array[index] == null)
            {
                array[index] = new LinkedList<HashTableItem<TKey, TValue>>();
            }

            HashTableItem<TKey, TValue> hashTable = new HashTableItem<TKey,
TValue>(key, value);

            LinkedListNode<HashTableItem<TKey, TValue>> nodeHashTable = new
LinkedListNode<HashTableItem<TKey, TValue>>(hashTable);

            array[index].AddFirst(nodeHashTable);

            return index;
        }


        public void Find(TKey key)
        {
            int index = Hash(key);

            if (array[index] == null)
            {
```

```
        Console.WriteLine("This elment no");
    }

    foreach (var item in array[index])
    {
        if (item.Key.Equals(key))
        {
           Console.WriteLine( array[index]);

        }
    }
}


public bool Delete(TKey key)
{
    int index = Hash(key);

    if (array[index] == null)
    {
        return false;
    }

    foreach (var item in array[index])
    {
        if (item.Key.Equals(key))
        {
            array[index].Remove(item);

            return true;
        }
    }

    return false;
}

public TValue GetValue(TKey key)
{
    int index = Hash(key);

    if (array[index] == null)
    {
        return default(TValue);
    }

    foreach (var item in array[index])
    {

        if (item.Key.Equals(key))
        {
            return item.Value;
        }

    }

    return default(TValue);

}

public void Show()
{
    foreach (var item in array)
    {
        if (item != null)
```

```
                {
                    foreach (var node in item)
                    {
                        Console.WriteLine("Key - {0},  value - {1}", node.Key,
node.Value);
                    }
                }
            }
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            int size = 252;
            HashTable<int, int> hashTable = new HashTable<int, int>(size);

            hashTable.Add(1, 1);
            hashTable.Add(1, 2);
            hashTable.Add(3, 5);
            hashTable.Add(4, 3);
            hashTable.Add(5, 4);
            hashTable.Add(8, 6);
            hashTable.Add(13, 7);

            hashTable.Show();

            Console.WriteLine(hashTable.Delete(13));


            hashTable.Show();

           Console.WriteLine( hashTable.GetValue(8));
            Console.ReadKey();
        }
    }
}
```