

①

```
public class Hash Table
```

```
{  
    String [ ][ ] table;  
    int table size;
```

```
    Hash Table ( int size)
```

```
{  
    table = new String [size][ ];  
    table size = size;
```

```
}
```

```
public void add ( String Key, String Value)
```

```
{  
    if (Key == null || Value == null)
```

```
{  
        System.out.println(" cannot be null ");
```

```
}
```

```
    int iter = 0;
```

```
    int code = Math.abs (Key.hashCode()) % table size;
```

```
    if (table [code] == null) table [code] = new String (Key, Value);
```

```
    else
```

```
{  
    while (table [code] != null)
```

```
{  
        if (table [code][0].equals (Key))
```

```
{  
            System.out.println(" Already submitted " + Key);
```

```
        if (iter == table size) return;
```

code++;

code % = table size

iter++

}

table[code] = new String[] { key, value }

public void remove(String key)

{ if (key == null) {out.println("Cannot input null value");

int iter = 0

int code = Math.abs(key.hashCode()) % table size;

if (table[code][0].equals(key))

table[code] = null;

else while (!table[code][0].equals(key))

{ if (iter == table size)

{out.println("cannot find word");

code++;

code % = table size

}

iter++;

table[code] = null;

public String get (String Key)

{

int iter = 0

int code = Math.abs (key.hashCode()) % tableSize

if (table [code] [0] .equals (key))
return table [code] [1];

else

{ while (! table [code] [0] .equals (key))

{ if (iter == tableSize) return "Could not return word";

code ++;

code %= tableSize;

iter ++;

}

return table [code] [1];

}

}

}