

```

#include <string>
using namespace std;

template <typename V>
class MapNode {
    public:
        string key;
        V value;
        MapNode* next;

        MapNode(string key, V value) {
            this->key = key;
            this->value = value;
            next = NULL;
        }

        ~MapNode() {
            delete next;
        }
};

template <typename V>
class ourmap {
    MapNode<V>** buckets;
    int count;
    int numBuckets;

    public:
    ourmap() {
        count = 0;
        numBuckets = 5;
        buckets = new MapNode<V>*[numBuckets];
        for (int i = 0; i < numBuckets; i++) {
            buckets[i] = NULL;
        }

    }

    ~ourmap() {
        for (int i = 0; i < numBuckets; i++) {
            delete buckets[i];
        }
        delete [] buckets;
    }

    int size() {
        return count;
    }
}

```

```

V getValue(string key) {

}

private:
int getBucketIndex(string key) {
    int hashCode = 0;

    int currentCoeff = 1;
    for (int i = key.length() - 1; i >= 0; i--) {
        hashCode += key[i] * currentCoeff;
        hashCode = hashCode % numBuckets;
        currentCoeff *= 37;
        currentCoeff = currentCoeff % numBuckets;
    }

    return hashCode % numBuckets;
}

public:
void insert(string key, V value) {
    int bucketIndex = getBucketIndex(string key);
    MapNode<V>* head = buckets[bucketIndex];
    while (head != NULL) {
        if (head->key == key) {
            head->value = value;
            return;
        }
        head = head->next;
    }
    head = buckets[bucketIndex];
    MapNode<V>* node = new MapNode<V>(key, value);
    node->next = head;
    buckets[bucketIndex] = node;
    count++;
}

};

```