#include <iostream>

#include <vector>

using namespace std;

class HashTable {

private:

    vector<int> table;

    int capacity;

    // Hash function (simple modulo)

    int hashFunction(int key) {

        return key % capacity;

    }

    // Probe function for linear probing

    int linearProbe(int key, int i) {

        return (hashFunction(key) + i) % capacity;

    }

public:

    HashTable(int size) : capacity(size) {

        table.resize(capacity, -1); // Initialize all slots to -1 (empty)

    }

    // Insert key into the hash table using linear probing

    void insert(int key) {

        int index = hashFunction(key);

        int i = 0;

        while (table[index] != -1) {

            index = linearProbe(key, ++i);

        }

        table[index] = key;

    }

    // Search for key in the hash table

    bool search(int key) {

        int index = hashFunction(key);

        int i = 0;

        while (table[index] != key) {

            if (table[index] == -1)

                return false; // Key not found

            index = linearProbe(key, ++i);

        }

        return true; // Key found

    }

    // Display the contents of the hash table

    void display() {

        cout << "Hash Table:" << endl;

        for (int i = 0; i < capacity; ++i) {

            if (table[i] != -1)

                cout << i << ": " << table[i] << endl;

            else

                cout << i << ": Empty" << endl;

        }

    }

};

int main() {

    int size, numKeys;

    cout << "Enter the size of the hash table: ";

    cin >> size;

    HashTable hashTable(size);

    cout << "Enter the number of keys to insert: ";

    cin >> numKeys;

    cout << "Enter the keys to insert:" << endl;

    for (int i = 0; i < numKeys; ++i) {

        int key;

        cin >> key;

        hashTable.insert(key);

    }

    cout << "Enter the key to search: ";

    int searchKey;

    cin >> searchKey;

    if (hashTable.search(searchKey)) {

        cout << "Key " << searchKey << " found in the hash table." << endl;

    } else {

        cout << "Key " << searchKey << " not found in the hash table." << endl;

    }

    hashTable.display();

    return 0;

}