```javascript
class HashTable {
  constructor(size) {
    this.values = {};
    this.numberOfValues = 0;
    this.size = size;
  }

  add(key, value) {
    const hash = this.calculateHash(key);
    if(!this.values.hasOwnProperty(hash)) {
      this.values[hash] = {};
    }
    if(!this.values[hash].hasOwnProperty(key)) {
      this.numberOfValues++;
    }
    this.values[hash][key] = value;
  }

  remove(key) {
    const hash = this.calculateHash(key);
    if(this.values.hasOwnProperty(hash) && this.values[hash].hasOwnProperty(key)) {
      delete this.values[hash][key];
      this.numberOfValues--;
    }
  }

  calculateHash(key) {
    return key.toString().length % this.size;
  }

  search(key) {
    const hash = this.calculateHash(key);
    if(this.values.hasOwnProperty(hash) && this.values[hash].hasOwnProperty(key)) {
      return this.values[hash][key];
    } else {
      return null;
    }
  }

  length() {
    return this.numberOfValues;
  }

  print() {
    let string = '';
    for(const value in this.values) {
      for(const key in this.values[value]) {
        string += `${this.values[value][key]} `;
      }
    }
    console.log(string.trim());
  }
}

const hashTable = new HashTable(3);
hashTable.add('first', 1);
hashTable.add('second', 2);
hashTable.add('third', 3);
hashTable.add('fourth', 4);
hashTable.add('fifth', 5);
hashTable.print(); // => 2 4 1 3 5
console.log('length gives 5:', hashTable.length()); // => 5
console.log('search second gives 2:', hashTable.search('second')); // => 2
hashTable.remove('fourth');
hashTable.remove('first');
hashTable.print(); // => 2 3 5
console.log('length gives 3:', hashTable.length()); // => 3

```