

```

#pragma once

#include <infos/util/map.h>
#include <arch/x86/pio.h>
#include <infos/util/string.h>

using namespace infos::util;

/**
 * class Node is a class used to create nodes of a dll . offset contains the offset provided in the read_block fn , we use isValid to bypass
the restriction
 * of not having a map.remove() , where although we dont remove the key from the map , we remove its node from the dll and make it invalid if
it exceeds
 * the given capacity and data is a pointer to the address which will contain the data to be cached.
 */

class Node {
public:
    int offset;
    uint8_t* data;
    bool isValid;
    Node* prev;
    Node* next;
};

class LRUCache {
public:
    LRUCache(int capacity);

    bool contains(int offset, void* buffer);

    void addLRU(int offset, void* buffer);

    void addMRU(int offset, void* buffer);

private:
    void makeMostRecent(Node* node);

    int _capacity;
    int _size;
    Map<int, Node*> _cache;
    Node* _head;
    Node* _tail;
};

```