```cpp
#include <bits/stdc++.h>
using namespace std;

class LRUCache {

    list<int> dq;
    unordered_map<int, list<int>::iterator> ma;
    int csize;

public:
    LRUCache(int);
    void refer(int);
    void display();
};
LRUCache::LRUCache(int n) { csize = n; }

void LRUCache::refer(int x)
{

    if (ma.find(x) == ma.end()) {

        if (dq.size() == csize) {

            int last = dq.back();


            dq.pop_back();


            ma.erase(last);
        }
    }


    else
        dq.erase(ma[x]);


    dq.push_front(x);
    ma[x] = dq.begin();
}

void LRUCache::display()
{


    for (auto it = dq.begin(); it != dq.end(); it++)
        cout << (*it) << " ";
    cout << endl;
}

int main()
{
    LRUCache ca(4);

    ca.refer(1);
    ca.refer(2);
    ca.refer(3);
    ca.refer(1);
    ca.refer(4);
    ca.refer(5);
    ca.display();

    return 0;
}
```