**#include <bits/stdc++.h>**

**using namespace std;**

**#define M 1000000007**

**class LRUCache {**

**private:**

**unordered\_map<int, int>keyValue;**

**list<int>l;**

**unordered\_map<int, list<int>::iterator>keyInListAddressMap;**

**int capacity;**

**int size;**

**public:**

**LRUCache(int capacity) {**

**this->capacity = capacity;**

**this->size = 0;**

**keyValue.reserve(capacity);**

**keyInListAddressMap.reserve(capacity);**

**}**

**int get(int key) {**

**if(this->keyValue.find(key) == this->keyValue.end()) {**

**return -1;**

**} else {**

**int value = this->keyValue[key];**

**l.erase(keyInListAddressMap[key]);**

**l.push\_front(key);**

**keyInListAddressMap[key] = l.begin();**

**return value;**

**}**

**}**

**void put(int key, int value) {**

**if(this->keyValue.find(key) == this->keyValue.end()) {**

**if(this->size == this->capacity) {**

**auto evictedKey = l.back();**

**keyValue.erase(evictedKey);**

**keyInListAddressMap.erase(evictedKey);**

**l.pop\_back();**

**} else {**

**this->size++;**

**}**

**} else {**

**l.erase(keyInListAddressMap[key]);**

**}**

**l.push\_front(key);**

**keyInListAddressMap[key] = l.begin();**

**keyValue[key] = value;**

**}**

**};**

**int main() {**

**int distinctCount = 0;**

**unordered\_map<int, int>mp;**

**int arr[]= {2, 2, 3, 1, 2,2 , 3, 1, 1, 4, 5, 1, 1};**

**int windowSize = 4;**

**for(int i=0;i<windowSize;i++){**

**if(mp[arr[i]] == 0 ) {**

**distinctCount++;**

**}**

**mp[arr[i]]++;**

**}**

**int n = 13;**

**mp.reserve(n);**

**int start = 0, end = windowSize;**

**vector<int>ans;**

**ans.push\_back(distinctCount);**

**while(end < n) {**

**if(mp[arr[start]] == 1) {**

**distinctCount--;**

**}**

**mp[arr[start++]]--;**

**if(mp[arr[end]] == 0) {**

**distinctCount++;**

**}**

**mp[arr[end++]]++;**

**ans.push\_back(distinctCount);**

**}**

**cout<<ans.size()<<endl;**

**for(int i=0;i<ans.size();i++){**

**cout<<ans[i]<<" ";**

**}**

**cout<<endl;**

**//**

**int N = 5, K = 3;**

**int arr1[] = {2, 3, 5, 4, 7};**

**unordered\_map<int, int>mp1;**

**for(int i=0;i<N;i++) {**

**if(arr1[i]>=K) {**

**mp1[arr1[i]]++;**

**}**

**}**

**int ans1 = 0;**

**for(int i=0;i<n;i++){**

**if(arr[i]>=K) {**

**int divFinder = arr[i] - K;**

**for(int j=1;j\*j<=divFinder;j++){**

**if(divFinder%j == 0){**

**if(mp1[j] > 0) {**

**ans1 += mp1[j];**

**}**

**if(mp1[divFinder/j] > 0) {**

**ans1 += mp1[j];**

**}**

**}**

**}**

**}**

**}**

**cout<<ans1<<endl;**

**return 0;**

**}**