

[Description](#)[My Submissions](#)[Hints/Editorial](#)[AC Submissions](#)[My Notes \(0\)](#)

? Ask Doubt

Time-Limit: 1 sec

Score: 0/100

Difficulty:

Memory: 256 MB

Accepted Submissions: 100

Relevant For: 

AZ-201

Description

You need to design and implement a data structure that is able to handle different types of queries. The data structure has a capacity **n** and at any moment it cannot store more than **n** keys. Implement the **LFUCache** class:

- 1. **LFUCache(int n):** Initializes the object with the capacity **n** of the data structure.
- 2. **int get(int key):** Gets the value of the key if the key exists in the cache. Otherwise, returns **-1**.
- 3. **void put(int key, int value):** Update the value of the key if present, or insert the key if not already present. When the cache reaches its capacity, it should invalidate and remove the **least frequently used** key before inserting a new item. For this problem, when there is a **tie** (i.e., two or more keys with the same frequency), the **least recently used** key would be invalidated.

To determine the least frequently used key, a **use counter** is maintained for each key in the cache. The key with the smallest **use counter** is the least frequently used key.

When a key is first inserted into the cache, its **use counter** is set to 1 (due to the put operation). The **use counter** for a key in the cache is incremented; either a **get** or **put** operation is called on it.

The **least recently used** key is the key with the smallest frequency which was called on the earliest among all the keys with the smallest frequency.

Input Format

Your LFUCache object will be instantiated and called as such:

```
LFUCache* obj = new LFUCache(n);
int param_1 = obj->get(key);
obj->put(key,value);
```

Output Format

For each **get** query the value of the key is printed and if the key is not present, '-1' is printed.

Constraints

```
1 <= n <= 10000
1 <= q <= 10000 (total number of function calls)
1 <= key <= 10000
0 <= value <= 10000
```

Sample Input 1

Copy



C++14[GCC] ▾

Submit

```
1  ▾ #include <bits/stdc++.h>
2    using namespace std;
3
4  ▾ class LFUCache {
```