**NAME** :- Manish Shashikant Jadhav

**UID** :- 2023301005.

**BRANCH** :- Comps -B. **BRANCH:** B.

**EXPERIMENT 8: Implement Page Replacement Algorithm.**

**SUBJECT** :- CAO (COMPUTER ARCHITECTURE AND ORGANIZATION)

**CODE :-**

#include <iostream>

#include <list>

#include <unordered\_map>

using namespace std;

class LRUCache {

private:

list<int> lruList;

unordered\_map<int, list<int>::iterator> pageMap;

int capacity;

public:

LRUCache(int capacity) {

this->capacity = capacity;

}

void referPage(int page) {

if (pageMap.find(page) == pageMap.end()) {

if (lruList.size() == capacity) {

int last = lruList.back();

lruList.pop\_back();

pageMap.erase(last);

}

} else {

lruList.erase(pageMap[page]);

}

lruList.push\_front(page);

pageMap[page] = lruList.begin();

}

void displayPages() {

for (auto it = lruList.begin(); it != lruList.end(); ++it)

cout << \*it << " ";

cout << endl;

}

};

int main() {

LRUCache cache(3);

cache.referPage(28);

cache.referPage(9);

cache.referPage(13);

cache.displayPages();

cache.referPage(2);

cache.displayPages();

cache.referPage(19);

cache.displayPages();

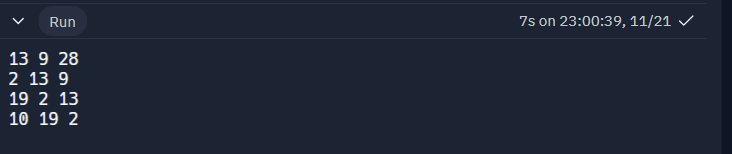
cache.referPage(10);

cache.displayPages();

return 0;

}

**OUTPUT** :-



**Conclusion:** Hence by completing this experiment I came to know about page replacement algorithm.