**Algorithm 9 :-** Write a program to demonstrate the use of LRU Page Replacement algorithm.

import java.util.\*;

class PageFrame {

int page;

long timestamp;

public PageFrame(int page, long timestamp) {

this.page = page;

this.timestamp = timestamp; } }

class LRU {

int capacity;

int size;

PageFrame[] frames;

public LRU(int capacity) {

this.capacity = capacity;

this.size = 0;

this.frames = new PageFrame[capacity]; }

public boolean isHit(int page) {

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

if (frames[i].page == page) {

return true; } }

return false; }

public void update(int page) {

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

if (frames[i].page == page) {

frames[i].timestamp = System.currentTimeMillis();

break; } } }

public void replace(int page) {

int minIndex = 0;

long minTimestamp = frames[0].timestamp;

for (int i = 1; i < size; i++) {

if (frames[i].timestamp < minTimestamp) {

minIndex = i;

minTimestamp = frames[i].timestamp; } }

frames[minIndex].page = page;

frames[minIndex].timestamp = System.currentTimeMillis(); }

public int process(int[] pages) {

int faults = 0;

for (int page : pages) {

if (isHit(page)) {

update(page); }

else {

faults++;

if (size < capacity) {

frames[size] = new PageFrame(page, System.currentTimeMillis());

size++; }

else {

replace(page); } } }

return faults; } }

class TestLRU {

public static void main(String[] args) {

Random random = new Random();

Scanner sc = new Scanner(System.in);

System.out.print("Enter no. of Frames : ");

int pn = sc.nextInt();

LRU lru = new LRU(pn);

int pages[] = new int[pn];

System.out.print("Pages : ");

for (int i = 0; i < pn; i++) {

pages[i] = random.nextInt(10);

System.out.print(pages[i] + " ");

}

System.out.println();

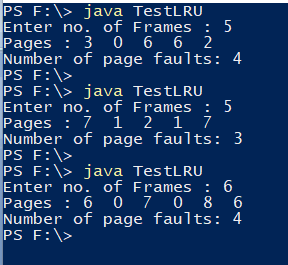
int faults = lru.process(pages);

System.out.println("Number of page faults: " + faults);

}

}

**OUTPUT :-**

****