**Algorithm 8 :-** Write a program to demonstrate the use of FIFO Page Replacement algorithm.

import java.util.HashSet;

import java.util.LinkedList;

import java.util.Queue;

import java.util.Random;

import java.util.Scanner;

class FIFO

{

static int pageFaults(int pages[], int n, int capacity) {

HashSet<Integer> s = new HashSet<>(capacity);

Queue<Integer> indexes = new LinkedList<>() ;

int page\_faults = 0;

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

if (s.size() < capacity) {

if (!s.contains(pages[i])) {

s.add(pages[i]);

page\_faults++;

indexes.add(pages[i]);

}

}

else {

if (!s.contains(pages[i])) {

int val = indexes.peek();

indexes.poll();

s.remove(val);

s.add(pages[i]);

indexes.add(pages[i]);

page\_faults++;

}

}

}

return page\_faults;

}

// Driver method

public static void main(String args[]) {

Random random = new Random();

Scanner sc = new Scanner(System.in);

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

int nf = sc.nextInt();

int pages[] = new int[nf];

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

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

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

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

}

int capacity = 4;

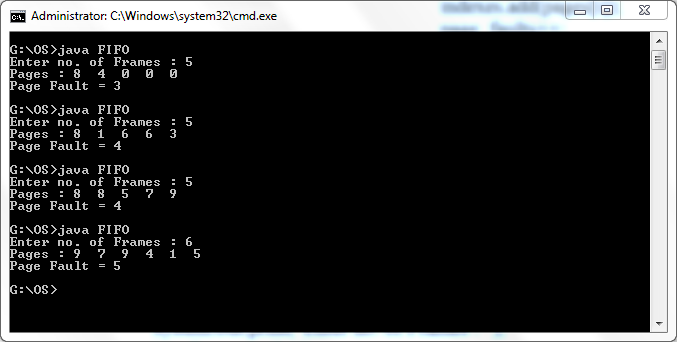
System.out.println();

System.out.println("Page Fault = " + pageFaults(pages, pages.length, capacity));

}

}

**OUTPUT :-**

****