Practical No.: 09

Aim: Write a program that implements the LRU page-replacement algorithm.

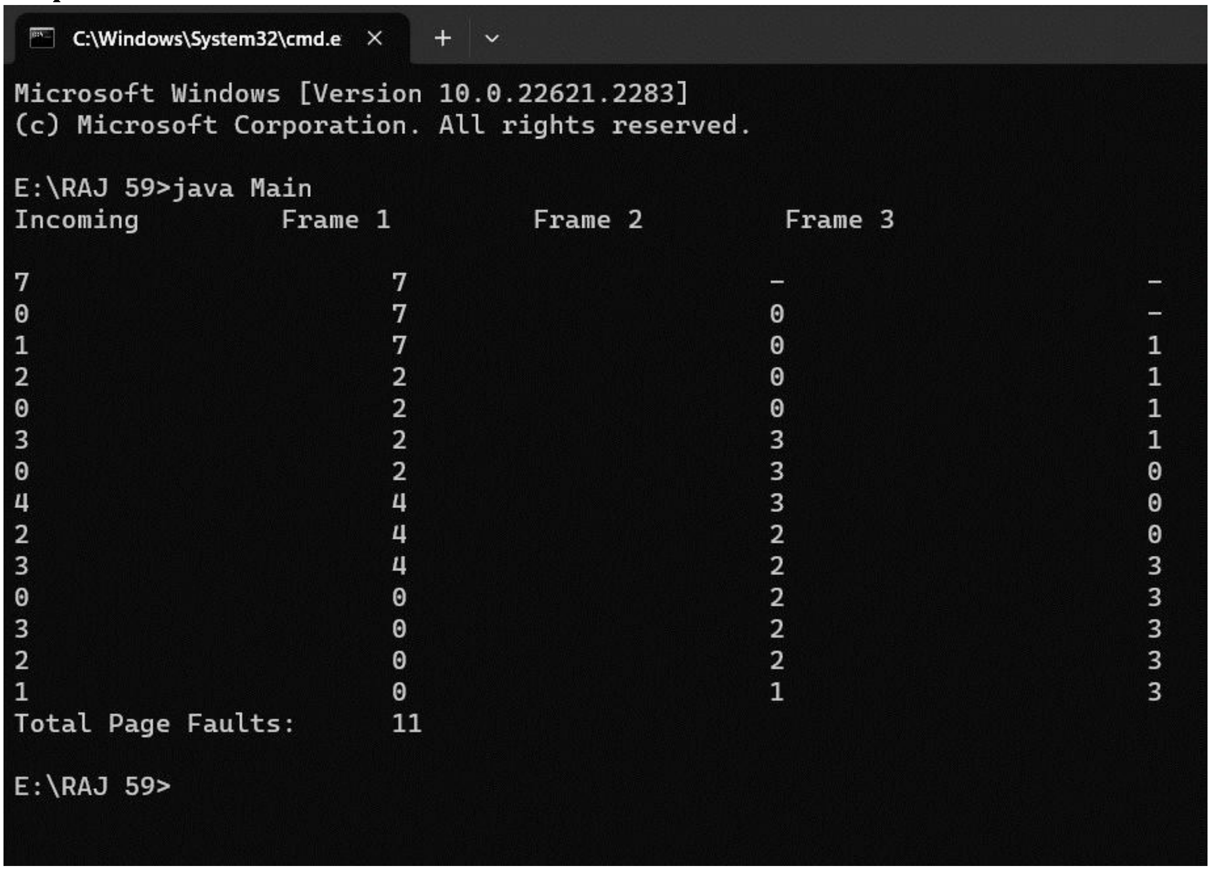
Source Code:

```
import java.util.Arrays;
public class Main{
	static boolean checkHit(int incomingPage, int[] queue, int occupied){
		for (int i =0; i < occupied; i++){
			if (incomingPage == queue[i])
			return true;
		}
		return false;
	}
	static void printFrame(int[] queue, int occupied){
		for (int i = 0; i < occupied; i++)
			System.out.print(queue[i] + "\t\t\t");
	}
	public static void main(String[] args){
		int[] incomingStream = {1,2,3,2,1,5,2,1,6,2,5,6,3,1,3};
		int n = incomingStream.length;
		int frames = 3;
		int[] queue = new int[frames];
		int[] distance = new int[frames];
		int occupied = 0;
		int pagefault = 0;
		System.out.println("Page\t Frame1 \t Frame2 \t Frame 3");
		for (int i = 0; i < n; i++){
			System.out.print(incomingStream[i] + ": \t\t");
			if (checkHit(incomingStream[i],queue,occupied)){
				printFrame(queue, occupied);
			}
			else if(occupied < frames)
			{
				queue[occupied] = incomingStream[i];
				pagefault++;
				occupied++;
				printFrame(queue,occupied);
			}
			else{
				int max = Integer.MIN_VALUE;
				int index = -1;
				for (int j = 0; j < frames; j++){
					distance[j] = 0;
					for (int k = i - 1; k >= 0; k--){
						++distance[j];
						if (queue[j] == incomingStream[k])
							break;
					}
```

```
                                        if (distance[j] > max){
                                                max = distance[j];
                                                index = j;
                                        }
                                }
                                queue[index] = incomingStream[i];
                                printFrame(queue,occupied);
                                pagefault++;
                        }
                        System.out.println();

                }
                System.out.println("Page Fault: " + pagefault);
        }
}
```

Output:



```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

E:\RAJ 59>java Main
Incoming         Frame 1         Frame 2         Frame 3

7                       7                       -                       -
0                       7                       0                       -
1                       7                       0                       1
2                       2                       0                       1
0                       2                       0                       1
3                       2                       3                       1
0                       2                       3                       0
4                       4                       3                       0
2                       4                       2                       0
3                       4                       2                       3
0                       0                       2                       3
3                       0                       2                       3
2                       0                       2                       3
1                       0                       1                       3
Total Page Faults:      11

E:\RAJ 59>
```