**Algorithm 10 :-** Write a program to demonstrate the use of Optimal Page Page Replacement algorithm.

import java.io.\*;

import java.util.\*;

class op {

static boolean search(int key, int[] fr) {

for (int i = 0; i < fr.length; i++)

if (fr[i] == key)

return true;

return false;

}

static int predict(int pg[], int[] fr, int pn, int index) {

int res = -1, farthest = index;

for (int i = 0; i < fr.length; i++) {

int j;

for (j = index; j < pn; j++) {

if (fr[i] == pg[j]) {

if (j > farthest) {

farthest = j;

res = i;

}

break;

}

}

if (j == pn)

return i;

}

return (res == -1) ? 0 : res;

}

static void optimalPage(int pg[], int pn, int fn) {

int[] fr = new int[fn];

int hit = 0;

int index = 0;

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

if (search(pg[i], fr)) {

hit++;

continue;

}

if (index < fn)

fr[index++] = pg[i];

else {

int j = predict(pg, fr, pn, i + 1);

fr[j] = pg[i];

}

}

System.out.println("No. of hits = " + hit);

System.out.println("No. of misses = " + (pn - hit));

}

// driver function

public static void main(String[] args)

{

Random random = new Random();

Scanner sc = new Scanner(System.in);

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

int fn = sc.nextInt();

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

int pn = sc.nextInt();

int pg[] = new int[pn];

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

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

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

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

}

System.out.println();

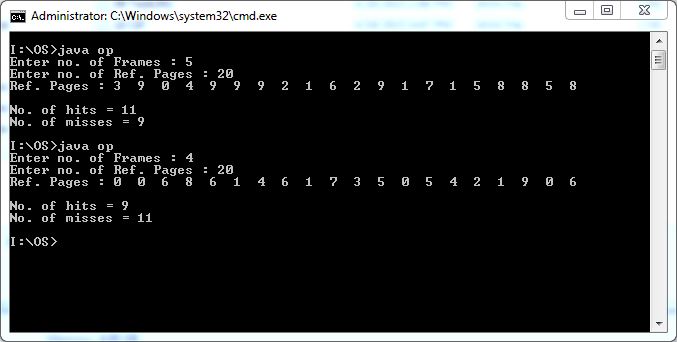
System.out.println();

optimalPage(pg, pn, fn);

}

}

**OUTPUT :-**

****