FIFO PAGE REPLACEMENT :  
import java.io.\*;  
public class FIFO {  
public static void main(String[] args) throws IOException  
{  
BufferedReader br = new BufferedReader(new  
InputStreamReader(System.in));  
int frames, pointer = 0, hit = 0, fault = 0,ref\_len;  
int buffer[];  
int reference[];  
int mem\_layout[][];  
System.out.println("Please enter the number of Frames: ");  
frames = Integer.parseInt(br.readLine());  
System.out.println("Please enter the length of the Reference  
string: ");  
ref\_len = Integer.parseInt(br.readLine());  
reference = new int[ref\_len];  
mem\_layout = new int[ref\_len][frames];  
buffer = new int[frames];  
for(int j = 0; j < frames; j++)  
buffer[j] = -1;  
System.out.println("Please enter the reference string: ");  
for(int i = 0; i < ref\_len; i++)  
{  
reference[i] = Integer.parseInt(br.readLine());  
}  
System.out.println();  
for(int i = 0; i < ref\_len; i++)  
{  
int search = -1;  
for(int j = 0; j < frames; j++)  
{  
if(buffer[j] == reference[i])  
{  
search = j;  
hit++;  
break;  
}  
}  
if(search == -1)  
{  
buffer[pointer] = reference[i];  
fault++;  
pointer++;  
if(pointer == frames)  
pointer = 0;  
}  
for(int j = 0; j < frames; j++)

mem\_layout[i][j] = buffer[j];  
}  
for(int i = 0; i < frames; i++)  
{  
for(int j = 0; j < ref\_len; j++)  
System.out.printf("%3d ",mem\_layout[j][i]);  
System.out.println();  
}  
System.out.println("The number of Hits: " + hit);  
System.out.println("Hit Ratio: " + (float)((float)hit/ref\_len));  
System.out.println("The number of Faults: " + fault);  
}  
}  
output:-  
Please enter the number of Frames:  
3  
Please enter the length of the Reference string:  
20  
Please enter the reference string:  
7  
0  
1  
2  
0  
3  
0  
4  
2  
3  
0  
3  
2  
1  
2  
0  
1  
7  
0  
1  
7 7 7 2 2 2 2 4 4 4 0 0 0 0 0 0 0 7  
7 7  
-1 0 0 0 0 3 3 3 2 2 2 2 2 1 1 1 1 1  
0 0  
-1 -1 1 1 1 1 0 0 0 3 3 3 3 3 2 2 2 2  
2 1  
The number of Hits: 5  
Hit Ratio: 0.25  
The number of Faults: 15  
--------------------------------

LRU Page Replacement algorithm in java  
code in Java:  
import java.io.\*;  
import java.util.\*;  
public class LRU {  
public static void main(String[] args) throws IOException  
{  
BufferedReader br = new BufferedReader(new  
InputStreamReader(System.in));  
int frames,pointer = 0, hit = 0, fault = 0,ref\_len;  
Boolean isFull = false;  
int buffer[];  
ArrayList<Integer> stack = new ArrayList<Integer>();  
int reference[];  
int mem\_layout[][];  
System.out.println("Please enter the number of Frames: ");  
frames = Integer.parseInt(br.readLine());  
System.out.println("Please enter the length of the Reference string:  
");  
ref\_len = Integer.parseInt(br.readLine());  
reference = new int[ref\_len];  
mem\_layout = new int[ref\_len][frames];  
buffer = new int[frames];  
for(int j = 0; j < frames; j++)  
buffer[j] = -1;  
System.out.println("Please enter the reference string: ");  
for(int i = 0; i < ref\_len; i++)  
{  
reference[i] = Integer.parseInt(br.readLine());  
}  
System.out.println();  
for(int i = 0; i < ref\_len; i++)  
{  
if(stack.contains(reference[i]))  
{  
stack.remove(stack.indexOf(reference[i]));  
}  
stack.add(reference[i]);  
int search = -1;  
for(int j = 0; j < frames; j++)  
{  
if(buffer[j] == reference[i])  
{  
search = j;  
hit++;  
break;

}  
}  
if(search == -1)  
{  
if(isFull)  
{  
int min\_loc = ref\_len;  
for(int j = 0; j < frames; j++)  
{  
if(stack.contains(buffer[j]))  
{  
int temp = stack.indexOf(buffer[j]);  
if(temp < min\_loc)  
{  
min\_loc = temp;  
pointer = j;  
}  
}  
}  
}  
buffer[pointer] = reference[i];  
fault++;  
pointer++;  
if(pointer == frames)  
{  
pointer = 0;  
isFull = true;  
}  
}  
for(int j = 0; j < frames; j++)  
mem\_layout[i][j] = buffer[j];  
}  
for(int i = 0; i < frames; i++)  
{  
for(int j = 0; j < ref\_len; j++)  
System.out.printf("%3d ",mem\_layout[j][i]);  
System.out.println();  
}  
System.out.println("The number of Hits: " + hit);  
System.out.println("Hit Ratio: " + (float)((float)hit/ref\_len));  
System.out.println("The number of Faults: " + fault);  
}  
}  
output:-  
Please enter the number of Frames:  
3  
Please enter the length of the Reference string:  
20  
Please enter the reference string:  
7  
0  
1  
2

0  
3  
0  
4  
2  
3  
0  
3  
2  
1  
2  
0  
1  
7  
0  
1  
7 7 7 2 2 2 2 4 4 4 0 0 0 1 1 1 1 1 1  
1  
-1 0 0 0 0 0 0 0 0 3 3 3 3 3 3 0 0 0 0  
0  
-1 -1 1 1 1 3 3 3 2 2 2 2 2 2 2 2 2 7 7  
7  
The number of Hits: 8  
Hit Ratio: 0.4  
The number of Faults: 12  
--------------------------------

Optimal Page Replacement algorithm in java  
code in Java:  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
public class OptimalReplacement {  
public static void main(String[] args) throws IOException  
{  
BufferedReader br = new BufferedReader(new  
InputStreamReader(System.in));  
int frames, pointer = 0, hit = 0, fault = 0,ref\_len;  
boolean isFull = false;  
int buffer[];  
int reference[];  
int mem\_layout[][];  
System.out.println("Please enter the number of Frames: ");  
frames = Integer.parseInt(br.readLine());  
System.out.println("Please enter the length of the Reference string:  
");  
ref\_len = Integer.parseInt(br.readLine());  
reference = new int[ref\_len];  
mem\_layout = new int[ref\_len][frames];  
buffer = new int[frames];  
for(int j = 0; j < frames; j++)  
buffer[j] = -1;  
System.out.println("Please enter the reference string: ");  
for(int i = 0; i < ref\_len; i++)  
{  
reference[i] = Integer.parseInt(br.readLine());  
}  
System.out.println();  
for(int i = 0; i < ref\_len; i++)  
{  
int search = -1;  
for(int j = 0; j < frames; j++)  
{  
if(buffer[j] == reference[i])  
{  
search = j;  
hit++;  
break;  
}  
}  
if(search == -1)  
{  
if(isFull)  
{  
int index[] = new int[frames];  
boolean index\_flag[] = new boolean[frames];

for(int j = i + 1; j < ref\_len; j++)  
{  
for(int k = 0; k < frames; k++)  
{  
if((reference[j] == buffer[k]) && (index\_flag[k] == false))  
{  
index[k] = j;  
index\_flag[k] = true;  
break;  
}  
}  
}  
int max = index[0];  
pointer = 0;  
if(max == 0)  
max = 200;  
for(int j = 0; j < frames; j++)  
{  
if(index[j] == 0)  
index[j] = 200;  
if(index[j] > max)  
{  
max = index[j];  
pointer = j;  
}  
}  
}  
buffer[pointer] = reference[i];  
fault++;  
if(!isFull)  
{  
pointer++;  
if(pointer == frames)  
{  
pointer = 0;  
isFull = true;  
}  
}  
}  
for(int j = 0; j < frames; j++)  
mem\_layout[i][j] = buffer[j];  
}  
for(int i = 0; i < frames; i++)  
{  
for(int j = 0; j < ref\_len; j++)  
System.out.printf("%3d ",mem\_layout[j][i]);  
System.out.println();  
}  
System.out.println("The number of Hits: " + hit);  
System.out.println("Hit Ratio: " + (float)((float)hit/ref\_len));  
System.out.println("The number of Faults: " + fault);  
}  
}  
output:-

/\*

Problem Statement :

Write a Java Program (using OOP features) to implement paging simulation using

1. Least Recently Used (LRU)

2. Optimal algorithm

\*\*\*\*Optimal\*\*\*\*

\*/

import java.util.\*;

import java.io.\*;

class Optimal

{

public static void main(String args[])throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int numberOfFrames, numberOfPages, flag1, flag2, flag3, i, j, k, pos = 0, max;

int faults = 0;

int temp[] = new int[10];

System.out.println("Enter number of Frames: ");

numberOfFrames = Integer.parseInt(br.readLine());

int frame[] = new int[numberOfFrames];

System.out.println("Enter number of Pages: ");

numberOfPages = Integer.parseInt(br.readLine());

int pages[] = new int[numberOfPages];

System.out.println("Enter the pages: ");

for(i=0; i<numberOfPages; i++)

pages[i] = Integer.parseInt(br.readLine());

for(i = 0; i < numberOfFrames; i++)

frame[i] = -1;

for(i = 0; i < numberOfPages; ++i){

flag1 = flag2 = 0;

for(j = 0; j < numberOfFrames; ++j){

if(frame[j] == pages[i]){

flag1 = flag2 = 1;

break;

}

}

if(flag1 == 0){

for(j = 0; j < numberOfFrames; ++j){

if(frame[j] == -1){

faults++;

frame[j] = pages[i];

flag2 = 1;

break;

}

}

}

if(flag2 == 0){

flag3 =0;

for(j = 0; j < numberOfFrames; ++j){

temp[j] = -1;

for(k = i + 1; k < numberOfPages; ++k){

if(frame[j] == pages[k]){

temp[j] = k;

break;

}

}

}

for(j = 0; j < numberOfFrames; ++j){

if(temp[j] == -1){

pos = j;

flag3 = 1;

break;

}

}

if(flag3 ==0){

max = temp[0];

pos = 0;

for(j = 1; j < numberOfFrames; ++j){

if(temp[j] > max){

max = temp[j];

pos = j;

}

}

}

frame[pos] = pages[i];

faults++;

}

// System.out.print();

for(j = 0; j < numberOfFrames; ++j){

System.out.print("\t"+ frame[j]);

}

}

System.out.println("\n\nTotal Page Faults: "+ faults);

}

}

//7 0 1 2 0 3 0 4 2 3 0 3 2

**/\***

**Problem Statement :**

**Write a Java Program (using OOP features) to implement paging simulation using**

**1. Least Recently Used (LRU)**

**2. Optimal algorithm**

**\*\*\*\*LRU\*\*\*\***

**\*/**

import java.io.\*;

class lru

{

public static void main(String args[])throws IOException

{

BufferedReader obj=new BufferedReader(new InputStreamReader(System.in));

int f,page=0,ch,pgf=0,n,chn=0;

boolean flag;

int pages[]; //pgf-page fault

System.out.println("1.LRU");

int pt=0;

System.out.println("enter no. of frames: ");

f=Integer.parseInt(obj.readLine());

int frame[]=new int[f];

for(int i=0;i<f;i++)

{

frame[i]=-1;

}

System.out.println("enter the no of pages ");

n=Integer.parseInt(obj.readLine());

pages=new int[n];

System.out.println("enter the page no ");

for(int j=0;j<n;j++)

pages[j]=Integer.parseInt(obj.readLine());

int pg=0;

for(pg=0;pg<n;pg++)

{

page=pages[pg];

flag=true;

for(int j=0;j<f;j++)

{

if(page==frame[j])

{

flag=false;

break;

}

}

int temp,h=3,i;

if(flag)

{

if( frame[1]!=-1 && frame[2]!=-1 && frame[0]!=-1)

{

temp=pages[pg-3];

if(temp==pages[pg-2] || temp==pages[pg-1])

temp=pages[pg-4];

for(i=0;i<f;i++)

if(temp==frame[i])

break;

frame[i]=pages[pg];

}

else

{

if(frame[0]==-1)

frame[0]=pages[pg];

else if(frame[1]==-1)

frame[1]=pages[pg];

else if(frame[2]==-1)

frame[2]=pages[pg];

}

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

for(int j=0;j<f;j++)

System.out.print(frame[j]+" ");

System.out.println();

pgf++;

}

else

{

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

for(int j=0;j<f;j++)

System.out.print(frame[j]+" ");

System.out.println();

}

}//for

System.out.println("Page fault:"+pgf);

}//main

}//class

/\*

OUTPUT:-

akshay@akshay-1011PX:~/Desktop/SPOS/LRU$ javac lru.java

akshay@akshay-1011PX:~/Desktop/SPOS/LRU$ java lru

1.LRU

enter no. of frames:

4

enter the no of pages

10

enter the page no

1

0

1

2

3

7

8

1

5

2

frame :1 -1 -1 -1

frame :1 0 -1 -1

frame :1 0 -1 -1

frame :1 0 2 -1

frame :1 3 2 -1

frame :7 3 2 -1

frame :7 3 8 -1

frame :7 1 8 -1

frame :5 1 8 -1

frame :5 1 2 -1

Page fault:9

akshay@akshay-1011PX:~/Desktop/SPOS/LRU$

\*/