**Group : B**

**Assignment No. : 7**

**Title : Write a program to implement Page Replacement Algorithms.**

**Roll No. : 2365**

**-------------------------------------------------------------------------------------------**

**import** java.util.\*;

**public** **class** PRAlgorithms

{

**public** **static** **void** main(String[] args)

{

Scanner scan=**new** Scanner(System.***in***);

**int** pages[]=**new** **int**[20];

**int** indexs[]=**new** **int**[20];

**int** pagefault=0;

**int** pagen;

System.***out***.println("\n\"PAGE DETAILS:\"");

System.***out***.println("-------------------");

System.***out***.println("Enter Number of pages ::");

pagen=scan.nextInt();

**for**(**int** i=0;i<pagen;i++)

{

System.***out***.println("\nEnter page no::");

pages[i]=scan.nextInt();

}

**int** ch;

**do**

{

System.***out***.println("\n----MENU----");

System.***out***.println("1.FIFO\n2.LRU\n3.OPTIMAL\n4.EXIT.");

System.***out***.println("Enter your choice::");

ch=scan.nextInt();

**switch**(ch)

{

**case** 1:

{

**int** frameno,count=0;

**int** j=0;

pagefault=0;

ArrayList<Integer> frames=**new** ArrayList(); //arraylist for frames

Queue<Integer> pageno=**new** LinkedList<Integer>(); //queue to add sequence of pages

System.***out***.println("\'FIFO PAGE REPLACEMENT ALGORITHM::");

System.***out***.println("\nEnter number of frames:: ");

frameno=scan.nextInt();

count=0;

**for**(**int** i=0;i<pagen;i++)

{

**if**(count!=frameno)

{

**if**(frames.isEmpty())

{

pageno.add(pages[i]);

frames.add(pages[i]);

count++;

pagefault++;

}

**else**

{

**if**(frames.contains(pages[i])==**true**)

{

System.***out***.println("\nPage " +pages[i]+" exists.");

}

**else**

{

pageno.add(pages[i]);

frames.add(pages[i]);

count++;

pagefault++;

}

}

}

**else**

{

**if**(frames.contains(pages[i])==**true**)

{

System.***out***.println("\nPage " +pages[i]+" exists.");

}

**else**

{

System.***out***.println("\nPage "+frames.get(j)+" is replaced by Page "+pages[i]+".");

frames.remove(j);

frames.add(j,pages[i]);

pagefault++;

j++;

**if**(j<=(frameno-1))

{

}

**else**

j=0;

}

}

**for**(**int** s=0;s<frames.size();s++)

{

System.***out***.println("|"+frames.get(s)+"|");

}

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

}

}

**break**;

**case** 2:

{

**int** frameno,count=0;

**int** j=0;

pagefault=0;

ArrayList<Integer> pageno=**new** ArrayList();

System.***out***.println("\'LRU PAGE REPLACEMENT ALGORITHM::");

System.***out***.println("\nEnter number of frames:: ");

frameno=scan.nextInt();

ArrayList<Integer> frames=**new** ArrayList(frameno); //arraylist for frames

count=0;

**for**(**int** i=0;i<pagen;i++)

{

**if**(count!=frameno)

{

**if**(frames.isEmpty())

{

pageno.add(pages[i]);

frames.add(pages[i]);

count++;

pagefault++;

}

**else**

{

**if**(frames.contains(pages[i])==**true**)

{

System.***out***.println("\nPage " +pages[i]+" exists.");

pageno.add(pages[i]);

}

**else**

{

pageno.add(pages[i]);

frames.add(pages[i]);

count++;

pagefault++;

}

}

}

**else**

{

**if**(frames.contains(pages[i])==**true**)

{

System.***out***.println("\nPage " +pages[i]+" exists.");

pageno.add(pages[i]);

}

**else**

{

**int** min=99;

**int** indexno[]=**new** **int**[6];

**int** rempg=0;

**for**(**int** k=0;k<frameno;k++)

{

indexno[k]=pageno.lastIndexOf(frames.get(k));

**if**(indexno[k]<min)

min=indexno[k];

rempg=pageno.get(min);

}

**int** index=frames.indexOf(rempg);

System.***out***.println("\nPage "+rempg+" is replaced by Page"+pages[i]+".");

frames.remove(index);

frames.add(index,pages[i]);

pageno.add(pages[i]);

pagefault++;

}

}

**for**(**int** s=0;s<frames.size();s++)

{

System.***out***.println("|"+frames.get(s)+"|");

}

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

}

}

**break**;

**case** 3:{

**int** frameno,count=0;

**int** j=0;

pagefault=0;

**int** flag=0,cnt=0,id=0;

ArrayList<Integer> wholepagearr=**new** ArrayList(pagen);

ArrayList<Integer> pageno=**new** ArrayList();

System.***out***.println("\'Optimal Page Replacement Algorithm::");

System.***out***.println("\nEnter number of frames:: ");

frameno=scan.nextInt();

ArrayList<Integer> frame=**new** ArrayList(frameno); //arraylist for frames

count=0;

**for**(**int** p=0;p<pagen;p++)

wholepagearr.add(p, (pages[p]));

**for**(**int** i=0;i<pagen;i++)

{

**if**(count!=frameno)

{

**if**(frame.isEmpty())

{

pageno.add(pages[i]);

frame.add(pages[i]);

count++;

pagefault++;

}

**else**

{

**if**(frame.contains(pages[i])==**true**)

{

System.***out***.println("\nPage " +pages[i]+" exists.");

pageno.add(pages[i]);

}

**else**

{

pageno.add(pages[i]);

frame.add(pages[i]);

count++;

pagefault++;

}

}

}

**else**

{

**if**(frame.contains(pages[i])==**true**)

{

System.***out***.println("\nPage " +pages[i]+" exists.");

pageno.add(pages[i]);

}

**else**

{

**int** max=0;

**int** indexno[]=**new** **int**[6];

**int** pageS[]=**new** **int**[6];

**int** rempage=0;

**for**(**int** k=0;k<frameno;k++)

{

indexno[k]=wholepagearr.lastIndexOf(frame.get(k));

**if**(i > indexno[k])

{

indexno[k]=99;

pages[k]=frame.get(k);

flag=1;

}

**else**

{

**if**(indexno[k]>max)

max=indexno[k];

rempage=wholepagearr.get(max);

}

}

**if**(flag!=1)

{

**int** index=frame.indexOf(rempage);

System.***out***.println("\nPage "+rempage+" is replaced by Page"+pages[i]+".");

frame.remove(index);

frame.add(index,pages[i]);

pageno.add(pages[i]);

pagefault++;

}

**else**

{

**for**(**int** p=0;p<frameno;p++)

{

**if**(indexno[p]==99)

{

cnt++;

id=p;

}

}

**if**(cnt==1)

{

rempage=frame.get(id);

**int** index=frame.indexOf(rempage);

System.***out***.println("\nPage "+rempage+" is replaced by Page"+pages[i]+".");

frame.remove(index);

frame.add(index,pages[i]);

pageno.add(pages[i]);

pagefault++;

}

**else**

{

**int** pg=0;**int** idd;

**int** min=99;

**while**(pg<frameno)

{

**if**(pages[pg]!=0)

{

idd=wholepagearr.indexOf(pages[pg]);

**if**(idd<min)

{

min=idd;

}

}

pg++;

}

rempage=wholepagearr.get(min); **int** index=frame.indexOf(rempage);

System.***out***.println("\nPage "+rempage+" is replaced by Page"+pages[i]+".");

frame.remove(index);

frame.add(index,pages[i]);

pageno.add(pages[i]);

pagefault++;

}

}

}

}

**for**(**int** s=0;s<frame.size();s++)

{

System.***out***.println("|"+frame.get(s)+"|");

}

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

}

}

**break**;

**case** 4:

System.***out***.println("----Thankyou----");

**break**;

**default**:

System.***out***.println("----Invalid choice!!----");

**break**;

}

}**while**(ch!=4);

}

}

/\*OUTPUT

"PAGE DETAILS:"

-------------------

Enter Number of pages ::

12

Enter page no::

1

Enter page no::

2

Enter page no::

3

Enter page no::

4

Enter page no::

1

Enter page no::

2

Enter page no::

5

Enter page no::

1

Enter page no::

2

Enter page no::

3

Enter page no::

4

Enter page no::

5

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

1

'FIFO PAGE REPLACEMENT ALGORITHM::

Enter number of frames::

3

|1|

Page fault::1

|1|

|2|

Page fault::2

|1|

|2|

|3|

Page fault::3

Page 1 is replaced by Page 4.

|4|

|2|

|3|

Page fault::4

Page 2 is replaced by Page 1.

|4|

|1|

|3|

Page fault::5

Page 3 is replaced by Page 2.

|4|

|1|

|2|

Page fault::6

Page 4 is replaced by Page 5.

|5|

|1|

|2|

Page fault::7

Page 1 exists.

|5|

|1|

|2|

Page fault::7

Page 2 exists.

|5|

|1|

|2|

Page fault::7

Page 1 is replaced by Page 3.

|5|

|3|

|2|

Page fault::8

Page 2 is replaced by Page 4.

|5|

|3|

|4|

Page fault::9

Page 5 exists.

|5|

|3|

|4|

Page fault::9

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

2

'LRU PAGE REPLACEMENT ALGORITHM::

Enter number of frames::

3

|1|

Page fault::1

|1|

|2|

Page fault::2

|1|

|2|

|3|

Page fault::3

Page 1 is replaced by Page4.

|4|

|2|

|3|

Page fault::4

Page 2 is replaced by Page1.

|4|

|1|

|3|

Page fault::5

Page 3 is replaced by Page2.

|4|

|1|

|2|

Page fault::6

Page 4 is replaced by Page5.

|5|

|1|

|2|

Page fault::7

Page 1 exists.

|5|

|1|

|2|

Page fault::7

Page 2 exists.

|5|

|1|

|2|

Page fault::7

Page 5 is replaced by Page3.

|3|

|1|

|2|

Page fault::8

Page 1 is replaced by Page4.

|3|

|4|

|2|

Page fault::9

Page 2 is replaced by Page5.

|3|

|4|

|5|

Page fault::10

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

3

'OPTIMAL PAGE REPLACEMENT ALGORITHM::

Enter number of frames::

3

Page 3 is replaced by Page4.

|1|

|2|

|4|

Page fault::4

Page 1 exists.

|1|

|2|

|4|

Page fault::4

Page 2 exists.

|1|

|2|

|4|

Page fault::4

Page 4 is replaced by Page5.

|1|

|2|

|5|

Page fault::5

Page 1 exists.

|1|

|2|

|5|

Page fault::5

Page 2 exists.

|1|

|2|

|5|

Page fault::5

Page 1 is replaced by Page3.

|3|

|2|

|5|

Page fault::6

Page 2 is replaced by Page4.

|3|

|4|

|5|

Page fault::7

Page 5 exists.

|3|

|4|

|5|

Page fault::7

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

4

----Thankyou----

OUTPUT 2:

"PAGE DETAILS:"

-------------------

Enter Number of pages ::

15

Enter page no::

1

Enter page no::

1

Enter page no::

2

Enter page no::

3

Enter page no::

4

Enter page no::

1

Enter page no::

1

Enter page no::

2

Enter page no::

3

Enter page no::

2

Enter page no::

4

Enter page no::

2

Enter page no::

2

Enter page no::

3

Enter page no::

3

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

1

'FIFO PAGE REPLACEMENT ALGORITHM::

Enter number of frames::

3

|1|

Page fault::1

Page 1 exists.

|1|

Page fault::1

|1|

|2|

Page fault::2

|1|

|2|

|3|

Page fault::3

Page 1 is replaced by Page 4.

|4|

|2|

|3|

Page fault::4

Page 2 is replaced by Page 1.

|4|

|1|

|3|

Page fault::5

Page 1 exists.

|4|

|1|

|3|

Page fault::5

Page 3 is replaced by Page 2.

|4|

|1|

|2|

Page fault::6

Page 4 is replaced by Page 3.

|3|

|1|

|2|

Page fault::7

Page 2 exists.

|3|

|1|

|2|

Page fault::7

Page 1 is replaced by Page 4.

|3|

|4|

|2|

Page fault::8

Page 2 exists.

|3|

|4|

|2|

Page fault::8

Page 2 exists.

|3|

|4|

|2|

Page fault::8

Page 3 exists.

|3|

|4|

|2|

Page fault::8

Page 3 exists.

|3|

|4|

|2|

Page fault::8

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

2

'LRU PAGE REPLACEMENT ALGORITHM::

Enter number of frames::

3

|1|

Page fault::1

Page 1 exists.

|1|

Page fault::1

|1|

|2|

Page fault::2

|1|

|2|

|3|

Page fault::3

Page 1 is replaced by Page4.

|4|

|2|

|3|

Page fault::4

Page 2 is replaced by Page1.

|4|

|1|

|3|

Page fault::5

Page 1 exists.

|4|

|1|

|3|

Page fault::5

Page 3 is replaced by Page2.

|4|

|1|

|2|

Page fault::6

Page 4 is replaced by Page3.

|3|

|1|

|2|

Page fault::7

Page 2 exists.

|3|

|1|

|2|

Page fault::7

Page 1 is replaced by Page4.

|3|

|4|

|2|

Page fault::8

Page 2 exists.

|3|

|4|

|2|

Page fault::8

Page 2 exists.

|3|

|4|

|2|

Page fault::8

Page 3 exists.

|3|

|4|

|2|

Page fault::8

Page 3 exists.

|3|

|4|

|2|

Page fault::8

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

3

'OPTIMAL PAGE REPLACEMENT ALGORITHM::

Enter number of frames::

3

|1|

Page fault::1

Page 1 exists.

|1|

Page fault::1

|1|

|2|

Page fault::2

|1|

|2|

|3|

Page fault::3

Page 3 is replaced by Page4.

|1|

|2|

|4|

Page fault::4

Page 1 exists.

|1|

|2|

|4|

Page fault::4

Page 1 exists.

|1|

|2|

|4|

Page fault::4

Page 2 exists.

|1|

|2|

|4|

Page fault::4

Page 1 is replaced by Page3.

|3|

|2|

|4|

Page fault::5

Page 2 exists.

|3|

|2|

|4|

Page fault::5

Page 4 exists.

|3|

|2|

|4|

Page fault::5

Page 2 exists.

|3|

|2|

|4|

Page fault::5

Page 2 exists.

|3|

|2|

|4|

Page fault::5

Page 3 exists.

|3|

|2|

|4|

Page fault::5

Page 3 exists.

|3|

|2|

|4|

Page fault::5

----MENU----

1.FIFO ALGORITHM.

2.LRU ALGORITHM

3.OPTIMAL ALGORITHM.

4.EXIT.

Enter your choice::

4

----Thankyou----

\*/