Name: Sanjivani Shivaji More

Roll no: 3238

Subject: SPOS

Assignment no:

FIFO:

```
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:
1
2
4
2
5
6
1
```

```
8

9

5

6

7

8

4

5

7

3

6

9

3


  1  1  1  1  5  5  5  8  8  8  6  6  6  4  4  4  3  3  3  3

 -1  2  2  2  2  6  6  6  9  9  9  7  7  7  5  5  5  6  6  6

 -1 -1  4  4  4  4  1  1  1  5  5  5  8  8  8  7  7  7  9  9

The number of Hits: 2

Hit Ratio: 0.1

The number of Faults: 18


Process finished with exit code 0
```

LRU:

```
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:
15
Please enter the reference string:
2
4
5
7
3
4
7
5
8
4
2
6
5
3
8

  2  2  2  7  7  7  7  7  7  4  4  4  5  5  5
 -1  4  4  4  3  3  3  5  5  5  2  2  2  3  3
 -1 -1  5  5  5  4  4  4  8  8  8  6  6  6  8
The number of Hits: 1
Hit Ratio: 0.06666667
The number of Faults: 14

Process finished with exit code 0
```

Optimal:

```
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:

```
Please enter the number of Frames:
3
Please enter the length of the Reference string:
14
Please enter the reference string:
2
5
7
3
6
8
4
9
3
8
9
7
6
6
  2   2   2   3   3   3   3   3   3   3   3   7   6   6
```

```
-1  5  5  5  6  8  8  8  8  8  8  8  8  8
-1 -1  7  7  7  7  4  9  9  9  9  9  9  9
The number of Hits: 4
Hit Ratio: 0.2857143
The number of Faults: 10

Process finished with exit code 0
```