

```

1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 public class OptimalReplacement {
5
6     public static void main(String[] args) throws IOException
7     {
8         BufferedReader br = new BufferedReader(new
9 InputStreamReader(System.in));
10        int frames, pointer = 0, hit = 0, fault = 0, ref_len;
11        boolean isFull = false;
12        int buffer[];
13        int reference[];
14        int mem_layout[][][];
15
16        System.out.println("Please enter the number of Frames: ");
17        frames = Integer.parseInt(br.readLine());
18
19        System.out.println("Please enter the length of the Reference string:
20 ");
21        ref_len = Integer.parseInt(br.readLine());
22
23        reference = new int[ref_len];
24        mem_layout = new int[ref_len][frames];
25        buffer = new int[frames];
26        for(int j = 0; j < frames; j++)
27            buffer[j] = -1;
28
29        System.out.println("Please enter the reference string: ");
30        for(int i = 0; i < ref_len; i++)
31        {
32            reference[i] = Integer.parseInt(br.readLine());
33        }
34        System.out.println();
35        for(int i = 0; i < ref_len; i++)
36        {
37            int search = -1;
38            for(int j = 0; j < frames; j++)
39            {
40                if(buffer[j] == reference[i])
41                {
42                    search = j;
43                    hit++;
44                    break;
45                }
46            }
47            if(search == -1)
48            {
49                if(isFull)
50                {
51                    int index[] = new int[frames];
52                    boolean index_flag[] = new boolean[frames];
53
54                    Made with Xodo PDF Reader and Editor
55                    for(int j = i + 1; j < ref_len; j++)
56                    {
57                        for(int k = 0; k < frames; k++)
58                        {
59                            if((reference[j] == buffer[k]) && (index_flag[k] == false))
60                            {
61                                index[k] = j;
62                                index_flag[k] = true;
63                                break;
64                            }
65                        }
66                    }
67                    int max = index[0];
68
69                    for(int k = 0; k < frames; k++)
70                    {
71                        if(index[k] > max)
72                        {
73                            max = index[k];
74                        }
75                    }
76
77                    for(int j = i + 1; j < ref_len; j++)
78                    {
79                        for(int k = 0; k < frames; k++)
80                        {
81                            if(index[k] == j)
82                            {
83                                buffer[k] = reference[j];
84                                index[k] = -1;
85                                index_flag[k] = false;
86                            }
87                        }
88                    }
89
90                    for(int k = 0; k < frames; k++)
91                    {
92                        if(index[k] == -1)
93                        {
94                            buffer[k] = reference[max];
95                            index[k] = max;
96                            index_flag[k] = true;
97                        }
98                    }
99
100                   isFull = true;
101
102                   System.out.println("Optimal Replacement Algorithm Output: ");
103                   for(int j = 0; j < ref_len; j++)
104                   {
105                       System.out.print(reference[j] + " ");
106                   }
107
108                   System.out.println();
109
110                   System.out.println("Number of hits: " + hit);
111                   System.out.println("Number of faults: " + fault);
112
113               }
114           }
115       }
116   }

```

```

67     pointer = 0;
68     if(max == 0)
69         max = 200;
70     for(int j = 0; j < frames; j++)
71     {
72         if(index[j] == 0)
73             index[j] = 200;
74         if(index[j] > max)
75             {
76             max = index[j];
77             pointer = j;
78         }
79     }
80 }
81 buffer[pointer] = reference[i];
82 fault++;
83 if(!isFull)
84 {
85     pointer++;
86     if(pointer == frames)
87     {
88         pointer = 0;
89         isFull = true;
90     }
91 }
92 }
93 for(int j = 0; j < frames; j++)
94     mem_layout[i][j] = buffer[j];
95 }

96 for(int i = 0; i < frames; i++)
97 {
98     for(int j = 0; j < ref_len; j++)
99         System.out.printf("%3d ",mem_layout[j][i]);
100    System.out.println();
101 }

102 System.out.println("The number of Hits: " + hit);
103 System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
104 System.out.println("The number of Faults: " + fault);
105 }
106 }
107 }
108 }
109 }
```