

```
(base) student@composnlab-93:~/Downloads/A43$ javac FIFO.java
(base) student@composnlab-93:~/Downloads/A43$ java FIFO
Please enter the number of Frames:
3
Please enter the length of the Reference string:
15
Please enter the reference string:
7
0
1
2
0
3
0
4
2
3
1
3
0
2
0
0
7 7 7 2 2 2 2 4 4 4 1 1 1 1 1
-1 0 0 0 0 3 3 3 2 2 2 2 0 0 0
-1 -1 1 1 1 1 0 0 0 3 3 3 3 2 2
The number of Hits: 3
Hit Ratio: 0.2
The number of Faults: 12
(base) student@composnlab-93:~/Downloads/A43$
```

```
1 import java.io.*;
2 public class FIFO {
3     public static void main(String[] args) throws IOException
4     {
5         BufferedReader br = new BufferedReader(new
6             InputStreamReader(System.in));
7         int frames, pointer = 0, hit = 0, fault = 0, ref_len;
8         int buffer[];
9         int reference[];
10        int mem_layout[][];
11        System.out.println("Please enter the number of Frames: ");
12        frames = Integer.parseInt(br.readLine());
13        System.out.println("Please enter the length of the Reference string: ");
14        ref_len = Integer.parseInt(br.readLine());
15        reference = new int[ref_len];
16        mem_layout = new int[ref_len][frames];
17        buffer = new int[frames];
18        for(int j = 0; j < frames; j++)
19            buffer[j] = -1;
20        System.out.println("Please enter the reference string: ");
21        for(int i = 0; i < ref_len; i++)
22        {
23            reference[i] = Integer.parseInt(br.readLine());
24        }
25        System.out.println();
26        for(int i = 0; i < ref_len; i++)
27        {
28            int search = -1;
29            for(int j = 0; j < frames; j++)
30            {
31                if(buffer[j] == reference[i])
32                {
33                    search = j;
34                    hit++;
35                    break;
36                }
37            }
38            if(search == -1)
39            {
40                buffer[pointer] = reference[i];
41                fault++;
42                pointer++;
43            }
44        }
45    }
46 }
```

```
42 pointer++;
43 if(pointer == frames)
44 pointer = 0;
45 }
46 for(int j = 0; j < frames; j++)
47 mem_layout[i][j] = buffer[j];
48 }
49 for(int i = 0; i < frames; i++)
50 {
51 for(int j = 0; j < ref_len; j++)
52 System.out.printf("%3d ", mem_layout[j][i]);
53 System.out.println();
54 }
55 System.out.println("The number of Hits: " + hit);
56 System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
57 System.out.println("The number of Faults: " + fault);
58 }
```

```
student@composnlab-93: ~/Downloads/A43
(base) student@composnlab-93:~/Downloads/A43$ javac OptimalReplacement.java
(base) student@composnlab-93:~/Downloads/A43$ java OptimalReplacement
Please enter the number of Frames:
4
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
7
3
3
3
3
3
3
3
3
1
1
1
1
1
1
1
1
-1
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
-1
-1
1
1
1
1
1
4
4
4
4
4
4
4
4
4
4
4
7
7
7
-1
-1
-1
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
The number of Hits: 12
Hit Ratio: 0.6
The number of Faults: 8
(base) student@composnlab-93:~/Downloads/A43$
```

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 public class OptimalReplacement {
5     public static void main(String[] args) throws IOException
6     {
7         BufferedReader br = new BufferedReader(new
8             InputStreamReader(System.in));
9         int frames, pointer = 0, hit = 0, fault = 0, ref_len;
10        boolean isFull = false;
11        int buffer[];
12        int reference[];
13        int mem_layout[][];
14        System.out.println("Please enter the number of Frames: ");
15        frames = Integer.parseInt(br.readLine());
16        System.out.println("Please enter the length of the Reference string: ");
17        ref_len = Integer.parseInt(br.readLine());
18        reference = new int[ref_len];
19        mem_layout = new int[ref_len][frames];
20        buffer = new int[frames];
21        for(int j = 0; j < frames; j++)
22            buffer[j] = -1;
23        System.out.println("Please enter the reference string: ");
24        for(int i = 0; i < ref_len; i++)
25        {
26            reference[i] = Integer.parseInt(br.readLine());
27        }
28        System.out.println();
29        for(int i = 0; i < ref_len; i++)
30        {
31            int search = -1;
32            for(int j = 0; j < frames; j++)
33            {
34                if(buffer[j] == reference[i])
35                {
36                    search = j;
37                    hit++;
38                    break;
39                }
40            }
41            if(search == -1)
42            {
43                fault++;
```

```
43 if(!isFull)
44 {
45     int index[] = new int[frames];
46     boolean index_flag[] = new boolean[frames];
47     for(int j = i + 1; j < ref_len; j++)
48     {
49         for(int k = 0; k < frames; k++)
50         {
51             if((reference[j] == buffer[k]) && (index_flag[k] == false))
52             {
53                 index[k] = j;
54                 index_flag[k] = true;
55                 break;
56             }
57         }
58     }
59     int max = index[0];
60     pointer = 0;
61     if(max == 0)
62         max = 200;
63     for(int j = 0; j < frames; j++)
64     {
65         if(index[j] == 0)
66             index[j] = 200;
67         if(index[j] > max)
68         {
69             max = index[j];
70             pointer = j;
71         }
72     }
73 }
74 buffer[pointer] = reference[i];
75 fault++;
76 if(!isFull)
```

```
76 if(!isFull)
77 {
78     pointer++;
79     if(pointer == frames)
80     {
81         pointer = 0;
82         isFull = true;
83     }
84 }
85 }
86 for(int j = 0; j < frames; j++)
87     mem_layout[i][j] = buffer[j];
88 }
89 for(int i = 0; i < frames; i++)
90 {
91     for(int j = 0; j < ref_len; j++)
92         System.out.printf("%3d ", mem_layout[j][i]);
93     System.out.println();
94 }
95 System.out.println("The number of Hits: " + hit);
96 System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
97 System.out.println("The number of Faults: " + fault);
98 }
```

```

-1  7  7  7  7  7  3  3  3  3  3  3  3  3  3  3  3  7  7  7
-1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
-1  -1  -1  1  1  1  1  4  4  4  4  4  4  1  1  1  1  1  1
-1  -1  -1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
The number of Hits: 12
Hlt Ratio: 0.6
The number of Faults: 8
(base) student@composnlab-93:~/Downloads/A43$

```



Open

LRU.java  
~/Downloads/A43

Save

```
1 import java.io.*;
2 import java.util.*;
3 public class LRU {
4     public static void main(String[] args) throws IOException
5     {
6         BufferedReader br = new BufferedReader(new
7             InputStreamReader(System.in));
8         int frames, pointer = 0, hit = 0, fault = 0, ref_len;
9         Boolean isFull = false;
10        int buffer[];
11        ArrayList<Integer> stack = new ArrayList<Integer>();
12        int reference[];
13        int mem_layout[][];
14        System.out.println("Please enter the number of Frames: ");
15        frames = Integer.parseInt(br.readLine());
16        System.out.println("Please enter the length of the Reference string: ");
17        ref_len = Integer.parseInt(br.readLine());
18        reference = new int[ref_len];
19        mem_layout = new int[ref_len][frames];
20        buffer = new int[frames];
21        for(int j = 0; j < frames; j++)
22            buffer[j] = -1;
23        System.out.println("Please enter the reference string: ");
24        for(int i = 0; i < ref_len; i++)
25        {
26            reference[i] = Integer.parseInt(br.readLine());
27        }
28        System.out.println();
29        for(int i = 0; i < ref_len; i++)
30        {
31            if(stack.contains(reference[i]))
32            {
33                stack.remove(stack.indexOf(reference[i]));
34            }
35            stack.add(reference[i]);
36            int search = -1;
37            for(int j = 0; j < frames; j++)
38            {
39                if(buffer[j] == reference[i])
40                {
41                    search = j;
42                    hit++;
43                    break;
44                }
45            }
46        }
47    }
48 }
```

Java Tab Width: 4 Ln 30, Col 2 INS

```
43 break;
44 }
45 }
46 if(search == -1)
47 {
48     if(isFull)
49     {
50         int min_loc = ref_len;
51         for(int j = 0; j < frames; j++)
52         {
53             if(stack.contains(buffer[j]))
54             {
55                 int temp = stack.indexOf(buffer[j]);
56                 if(temp < min_loc)
57                 {
58                     min_loc = temp;
59                     pointer = j;
60                 }
61             }
62         }
63     }
64     buffer[pointer] = reference[i];
65     fault++;
66     pointer++;
67     if(pointer == frames)
68     {
69         pointer = 0;
70         isFull = true;
71     }
72 }
73 for(int j = 0; j < frames; j++)
74     mem_layout[i][j] = buffer[j];
75 }
76 for(int i = 0; i < frames; i++)
77 {
78     for(int j = 0; j < ref_len; j++)
79         System.out.printf("%3d ", mem_layout[j][i]);
80     System.out.println();
81 }
82 System.out.println("The number of Hits: " + hit);
83 System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
84 System.out.println("The number of Faults: " + fault);
85 }
```