```
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)
hit / ref_len);
    System.out.println("The number of
Faults: " + fault);
```

```
    }
}
```

```
java -cp /tmp/QZ3hT2wnXB/OptimalReplacement
Please enter the number of Frames:
3
Please enter the length of the Reference string:
7
Please enter the reference string:
5
4
1
3
7
6
3

  5   5   5   3   3   3   3
 -1   4   4   4   7   6   6
 -1  -1   1   1   1   1   1
The number of Hits: 1
Hit Ratio: 0.14285715
The number of Faults: 6

=== Code Execution Successful ===
```