```
import java.util.*;

class LruAlgo {
    int[] p, fr, fs;   // Arrays for page reference, frame, and status
    int n, m, index, flag1 = 0, flag2 = 0, pf = 0, frsize = 3, i, j;

    Scanner src = new Scanner(System.in);
    void read() {
        System.out.println("Enter page table size");
        n = src.nextInt();
        p = new int[n];
        System.out.println("Enter elements in page table");
        for (int i = 0; i < n; i++)
            p[i] = src.nextInt();

        System.out.println("Enter page frame size");
        m = src.nextInt();
        fr = new int[m];
        fs = new int[m];
    }
    void display() {
        System.out.println("\n");
        for (i = 0; i < m; i++) {
            if (fr[i] == -1)
                System.out.print("[ ] ");
            else
                System.out.print("[" + fr[i] + "] ");
        }
        System.out.println();
    }
    void lru() {
        for (i = 0; i < m; i++) {
```

```
    fr[i] = -1;  // Initialize the page frames to -1
}
for (j = 0; j < n; j++) {
    flag1 = 0;
    flag2 = 0;

    // Check if the page is already in one of the frames
    for (i = 0; i < m; i++) {
        if (fr[i] == p[j]) {
            flag1 = 1;
            flag2 = 1;
            break;
        }
    }
    if (flag1 == 0) {
        // Check if there is any empty frame
        for (i = 0; i < m; i++) {
            if (fr[i] == -1) {
                fr[i] = p[j];
                flag2 = 1;
                pf++;
                break;
            }
        }
    }
    if (flag2 == 0) {
        // If no empty frame and page not found, we apply the LRU replacement
        Arrays.fill(fs, 0);

        // Mark the recently used pages in the frame
        for (int k = j - 1, l = 1; l <= frsize - 1 && k >= 0; l++, k--) {
            for (i = 0; i < m; i++) {
```

```
                if (fr[i] == p[k]) {
                    fs[i] = 1;  // Mark as recently used
                }
            }
        }
        // Find the least recently used page (unmarked in fs)
        for (i = 0; i < m; i++) {
            if (fs[i] == 0) {
                index = i;
                break;
            }
        }
        fr[index] = p[j];  // Replace with new page
        pf++;  // Increment page fault counter
      }

      System.out.print("Page : " + p[j]);
      display();
    }
    System.out.println("\nNumber of page faults: " + pf);
  }

  public static void main(String[] args) {
    LruAlgo a = new LruAlgo();
    a.read();
    a.lru();
  }
}
```

```
java -cp /tmp/WiR67IrZrM/LruAlgo
Enter page table size
3
Enter elements in page table
4
5
7
Enter page frame size
2
Page : 4

[4] [ ]
Page : 5

[4] [5]
Page : 7

[7] [5]

Number of page faults: 3

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