#include <iostream>

using namespace std;

int main() {

int pages[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1};

int frameSize = 3;

int n = sizeof(pages) / sizeof(pages[0]);

int frame[frameSize] = {-1, -1, -1}; // Initialize frame with -1 (empty)

int pageFaults = 0;

for (int i = 0; i < n; i++) {

bool pageFound = false;

// Check if page is already in frame (hit)

for (int j = 0; j < frameSize; j++) {

if (frame[j] == pages[i]) {

pageFound = true;

cout << "Page hit: " << pages[i] << endl;

break;

}

}

// If page not found, handle page fault

if (!pageFound) {

int lruIndex = 0;

// Find LRU index (least recently used)

for (int j = 1; j < frameSize; j++) {

if (frame[j] == -1 || frame[j] == pages[i]) {

lruIndex = j;

break;

}

}

// Replace LRU or empty slot

frame[lruIndex] = pages[i];

pageFaults++;

cout << "Page fault: " << pages[i] << endl;

}

}

cout << "Total Page Faults: " << pageFaults << endl;

return 0;

}