// Program to calculate number of page faults for reference string for LRU page replacement algorithm

#include <stdio.h>

#define MAX\_FRAMES 10

int findLRU(int frames[], int time[], int frameCount) {

int min = time[0], minIndex = 0;

for (int i = 1; i < frameCount; i++) {

if (time[i] < min) {

min = time[i];

minIndex = i;

}

}

return minIndex; // Return the index of the LRU page

}

int isPageInFrames(int frames[], int frameCount, int page) {

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

if (frames[i] == page) {

return 1; // Page found in frames

}

}

return 0; // Page not found

}

int main() {

int frameCount, pageCount;

int pageFaults = 0;

// Input: Number of frames and number of pages in the reference string

printf("Enter the number of frames: ");

scanf("%d", &frameCount);

printf("Enter the number of pages in the reference string: ");

scanf("%d", &pageCount);

int pages[pageCount];

printf("Enter the reference string (space-separated): ");

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

scanf("%d", &pages[i]);

}

int frames[MAX\_FRAMES];

int time[MAX\_FRAMES]; // Array to keep track of the last used time of each frame

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

frames[i] = -1; // Initialize frames as empty

time[i] = 0; // Initialize the last used time

}

// Processing each page in the reference string

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

int currentPage = pages[i];

// Check if the current page is already in the frames

if (!isPageInFrames(frames, frameCount, currentPage)) {

// Page fault occurs as the page is not in frames

int lruIndex = findLRU(frames, time, frameCount); // Find the index of the LRU page

frames[lruIndex] = currentPage; // Replace the LRU page with the current page

pageFaults++; // Increment page faults

}

// Update the time of the current page

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

if (frames[j] == currentPage) {

time[j] = i; // Update the last used time for the current page

break;

}

}

}

printf("\nTotal Page Faults: %d\n", pageFaults);

return 0;

}