14.page faults

#include <stdio.h>

#include <stdbool.h>

#define MAX\_PAGES 100

int pageFrames[MAX\_PAGES];

int numPageFrames = 0;

int pageFaults = 0;

bool isPageInFrames(int page)

{

int i;

for (i = 0; i < numPageFrames; i++) {

if (pageFrames[i] == page) {

return true;

}

}

return false;

}

int getPageToReplace(int pageReferences[], int startIndex, int numReferences)

{

int pageToReplace = -1;

int maxFutureDistance = -1;

int i, j;

for (i = 0; i < numPageFrames; i++) {

int futureDistance = -1;

for (j = startIndex; j < numReferences; j++) {

if (pageFrames[i] == pageReferences[j]) {

futureDistance = j - startIndex;

break;

}

}

if (futureDistance == -1) {

return i; // Found an empty frame, replace it

}

if (futureDistance > maxFutureDistance) {

maxFutureDistance = futureDistance;

pageToReplace = i;

}

}

return pageToReplace;

}

void displayPageFrames()

{

int i;

printf("Page frames: ");

for (i = 0; i < numPageFrames; i++) {

printf("%d ", pageFrames[i]);

}

printf("\n");

}

void simulatePageReferences(int pageReferences[], int numReferences)

{

int i;

for (i = 0; i < numReferences; i++) {

if (!isPageInFrames(pageReferences[i])) {

int pageToReplace = getPageToReplace(pageReferences, i+1, numReferences);

pageFrames[pageToReplace] = pageReferences[i];

pageFaults++;

}

displayPageFrames();

}

}

int main()

{

int numPageFrames, numPageReferences, i;

int pageReferences[MAX\_PAGES];

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

scanf("%d", &numPageFrames);

printf("Enter the number of page references: ");

scanf("%d", &numPageReferences);

printf("Enter the page reference sequence: ");

for (i = 0; i < numPageReferences; i++) {

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

}

for (i = 0; i < numPageFrames; i++) {

pageFrames[i] = -1;

}

simulatePageReferences(pageReferences, numPageReferences);

printf("Number of page faults: %d\n", pageFaults);

return 0;

}

OUTPUT

