/\*

# PROGRAM TO IMPLEMENT PAGE REPLACEMENT ALGORITHMS - LRU

Ashis Solomon

CS4B 17

MDL20CS035

\*/

**CODE:**

#include <stdio.h>

int findLRU(int time[], int fCount) {

int k, min, pos;

pos = 0;

min = time[0];

for (k = 1; k < fCount; ++k) {

if (time[k] < min) {

min = time[k];

pos = k;

}

}

return pos;

}

void LRU(int pages[], int frames[], int time[], int fC, int pC) {

printf("\nRef.String |\tFrames\n");

printf("-------------------------------\n");

int i, j, k, pos, flag, faultCount, counter, queue;

counter = 0, queue = 0, faultCount = 0;

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

flag = 0;

printf(" %d\t\t\t|\t", pages[i]);

for (j = 0; j < fC; ++j) {

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

flag = 1;

counter++;

time[j] = counter;

printf(" Hit\n\n");

break;

}

}

if ((flag == 0) && (queue < fC)) {

faultCount++;

counter++;

frames[queue] = pages[i];

time[queue] = counter;

queue++;

} else if ((flag == 0) && (queue == fC)) {

faultCount++;

counter++;

pos = findLRU(time, fC);

frames[pos] = pages[i];

time[pos] = counter;

}

if (flag == 0) {

for (k = 0; k < fC; ++k) {

printf("%d ", frames[k]);

}

printf("\n\n");

}

}

printf("\n Total Page Faults = %d\n\n", faultCount);

}

int main() {

int i, pC, fC, pages[30], frames[20], time[20];

printf("\n LRU \n");

printf("\n Number of Frames : ");

scanf("%d", & fC);

for (i = 0; i < fC; ++i)

frames[i] = -1;

printf("\n Number of Pages : ");

scanf("%d", & pC);

printf("\n Enter the reference string : ");

for (i = 0; i < pC; ++i)

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

LRU(pages, frames, time, fC, pC);

return 0;

}

**OUTPUT:**

