**Name : Keyur Patel**

**Roll No : 16010421073**

**Batch : A3**

**Exp-no : 7**

**Date: 21-10-23**

Code:

#include <stdio.h>

#include <stdlib.h>

#define MAX\_FRAMES 3

// Structure to represent a page

typedef struct {

int page\_number;

int time;

} Page;

// Structure to represent a frame

typedef struct {

int frame\_number;

Page page;

} Frame;

Frame frames[MAX\_FRAMES];

void initializeFrames() {

for (int i = 0; i < MAX\_FRAMES; i++) {

frames[i].frame\_number = i;

frames[i].page.page\_number = -1;

frames[i].page.time = -1;

}

}

void displayFrames() {

printf("Frames: ");

for (int i = 0; i < MAX\_FRAMES; i++) {

if (frames[i].page.page\_number != -1) {

printf("%d ", frames[i].page.page\_number);

} else {

printf("empty ");

}

}

printf("\n");

}

int findLRUFrame() {

int min\_time = frames[0].page.time;

int lru\_frame = 0;

for (int i = 1; i < MAX\_FRAMES; i++) {

if (frames[i].page.time < min\_time) {

min\_time = frames[i].page.time;

lru\_frame = i;

}

}

return lru\_frame;

}

int main() {

initializeFrames();

int pages[] = {1, 2, 3, 4, 1, 3, 5, 4, 6, 7, 8, 2, 1};

int page\_count = sizeof(pages) / sizeof(pages[0]);

int page\_index = 0;

for (int i = 0; i < page\_count; i++) {

int page\_number = pages[i];

displayFrames();

// Check if the page is already in a frame

int found = 0;

for (int j = 0; j < MAX\_FRAMES; j++) {

if (frames[j].page.page\_number == page\_number) {

frames[j].page.time = i;

found = 1;

break;

}

}

// If the page is not found in any frame, perform replacement

if (!found) {

int lru\_frame = findLRUFrame();

frames[lru\_frame].page.page\_number = page\_number;

frames[lru\_frame].page.time = i;

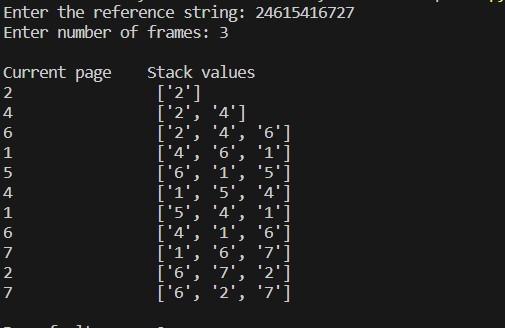
}

}

return 0;

}

Output:



# Outcomes:

**CO3:** Understand I/O management, memory management and file management

# Conclusion: (Conclusion to be based on outcomes achieved)

Through this experiment we implemented the Least Recently Used (LRU) page replacement algorithm. We calculated number of hits and miss and also calculated their ratios.