**EX 31: Construct a C program to simulate the First in First Out paging technique of memory management.**

**Aim**

To construct a C program that simulates the **First In First Out (FIFO)** paging technique for page replacement in memory management.

**Algorithm**

1. **Input:**
   * Get the number of frames (memory slots) available.
   * Get the sequence of page references.
2. **Process:**
   * Use a queue (array) to represent the frames.
   * If a page is not in the frames (page fault), replace the oldest page in the queue.
   * Update the page fault count for every page fault.
3. **Output:**
   * Display the frame contents for each step.
   * Print the total number of page faults.

**Program:**

#include <stdio.h>

#include <stdbool.h>

#define MAX\_FRAMES 10

#define MAX\_REFERENCES 30

// Function to check if a page is already in the frames

bool isPageInFrame(int frames[], int frame\_count, int page) {

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

if (frames[i] == page) {

return true;

}

}

return false;

}

// Function to print the current state of the frames

void printFrames(int frames[], int frame\_count) {

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

if (frames[i] == -1) {

printf(" - ");

} else {

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

}

}

printf("\n");

}

int main() {

int frame\_count, reference\_count;

int frames[MAX\_FRAMES], reference\_sequence[MAX\_REFERENCES];

int page\_faults = 0;

int pointer = 0; // Points to the next frame to replace

// Input: Number of frames

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

scanf("%d", &frame\_count);

// Initialize frames

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

frames[i] = -1; // -1 indicates an empty frame

}

// Input: Page reference sequence

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

scanf("%d", &reference\_count);

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

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

scanf("%d", &reference\_sequence[i]);

}

// FIFO page replacement algorithm

printf("\nPage Reference | Frames\n");

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

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

int current\_page = reference\_sequence[i];

printf(" %d |", current\_page);

// Check if the page is already in the frames

if (!isPageInFrame(frames, frame\_count, current\_page)) {

// Page fault occurs

frames[pointer] = current\_page; // Replace the oldest page

pointer = (pointer + 1) % frame\_count; // Update pointer

page\_faults++;

}

// Print the current state of frames

printFrames(frames, frame\_count);

}

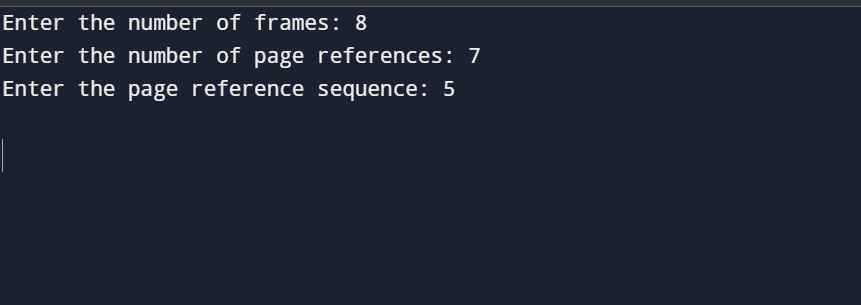
// Output: Total page faults

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

printf("Total Page Faults: %d\n", page\_faults);

return 0;

**OUTPUT:**

****