

Name: Vaibhav Kumar Jonwal

Roll no.: 202251150

---

## Paging technique in c. code

```
#include <stdio.h>
#include <stdlib.h>

// Constants for memory configuration
#define FRAME_SIZE 256 // Each frame is 256 bytes
#define TOTAL_FRAMES 8 // Total of 8 frames in memory
#define TOTAL_PAGES 16 // Total of 16 pages

// Page table to map pages to frames
int pageMapping[TOTAL_PAGES];
char memory[TOTAL_FRAMES][FRAME_SIZE];

// Initialize the page table with default values
void initializePageMapping() {
    for (int i = 0; i < TOTAL_PAGES; i++) {
        pageMapping[i] = -1; // -1 indicates the page is not mapped
    }
}

// Load a specific page into a specified frame
void loadPageIntoFrame(int pageNumber, int frameNumber) {
    if (pageNumber ≥ TOTAL_PAGES) {
        printf("Error: Invalid page number.\n");
        return;
    }
    pageMapping[pageNumber] = frameNumber;

    // Populate the frame with page content (for simulation)
    for (int i = 0; i < FRAME_SIZE; i++) {
        memory[frameNumber][i] = 'A' + pageNumber;
    }
    printf("Loaded page %d into frame %d.\n", pageNumber, frameNumber);
}

// Display the current state of the page table
void printPageMapping() {
    printf("Page Table:\n");
    for (int i = 0; i < TOTAL_PAGES; i++) {
        if (pageMapping[i] ≠ -1) {
            printf("Page %d → Frame %d\n", i, pageMapping[i]);
        } else {

```

```

        printf("Page %d → Not Loaded\n", i);
    }
}

// Access a page and display its corresponding content
void accessPageContent(int pageNumber) {
    if (pageNumber ≥ TOTAL_PAGES) {
        printf("Error: Invalid page number.\n");
        return;
    }

    int frameNumber = pageMapping[pageNumber];
    if (frameNumber == -1) {
        printf("Page %d is not currently loaded in memory.\n", pageNumber);
    } else {
        printf("Accessing page %d in frame %d: First byte contains '%c'\n",
            pageNumber, frameNumber, memory[frameNumber][0]);
    }
}

int main() {
    initializePageMapping();

    // Load a few pages into frames for testing
    loadPageIntoFrame(0, 0);
    loadPageIntoFrame(1, 1);
    loadPageIntoFrame(2, 2);
    loadPageIntoFrame(3, 3);

    // Display the current page table
    printPageMapping();

    // Access some pages and display their content
    accessPageContent(0);
    accessPageContent(2);
    accessPageContent(4); // Attempting to access a page not loaded

    return 0;
}

```

---

## Output

```

Loaded page 0 into frame 0.
Loaded page 1 into frame 1.
Loaded page 2 into frame 2.
Loaded page 3 into frame 3.
Page Table:
Page 0 → Frame 0
Page 1 → Frame 1
Page 2 → Frame 2
Page 3 → Frame 3
Page 4 → Not Loaded

```

Page 5 → Not Loaded  
Page 6 → Not Loaded  
Page 7 → Not Loaded  
Page 8 → Not Loaded  
Page 9 → Not Loaded  
Page 10 → Not Loaded  
Page 11 → Not Loaded  
Page 12 → Not Loaded  
Page 13 → Not Loaded  
Page 14 → Not Loaded  
Page 15 → Not Loaded  
Accessing page 0 in frame 0: First byte contains 'A'  
Accessing page 2 in frame 2: First byte contains 'C'  
Page 4 is not currently loaded in memory.