Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Computer Technology B. Tech in Computer Science and Engineering (IOT)

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Session 2025-2026

Vision To provide quality education in computer science and engineering that fosters innovation, technical excellence, and research, preparing students to meet global challenges in technology and society.	**Mission:** The mission is to impart strong fundamental knowledge of computing and operating system concepts while promoting problem-solving, analytical thinking, and hands-on learning through practical implementation. It also aims to encourage teamwork, ethical values, and lifelong learning for professional growth, preparing students to contribute effectively to technological advancement and meet industry needs.

Program Educational Objectives of the program (PEO): (broad statements that describe the professional and career accomplishments)

PEO1	**Preparation**	**P: Preparation**	**Pep-CL abbreviation pronounce as Pep-si-lL easy to recall**
PEO2	**Core Competence**	**E: Environment (Learning Environment)**	
PEO3	**Breadth**	**P: Professionalism**	
PEO4	**Professionalism**	**C: Core Competence**	
PEO5	**Learning Environment**	**L: Breadth (Learning in diverse areas)**	

Program Outcomes (PO): (statements that describe what a student should be able to do and know by the end of a program)

Keywords of POs:
Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

PSO Keywords: Cutting edge technologies, Research

"I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life." *to contribute to the development of cutting-edge technologies and Research*.

Integrity: I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

Name and Signature of Student and Date
(Signature and Date in Handwritten)

Session	**2025-26 (ODD)**	**Course Name**	**Operating System**
Semester	**5**	**Course Code**	**23IOT1504**
Roll No	**42**	**Name of Student**	**Karan F. Chopkar**

Practical Number	**6**
Course Outcome	CO4: After performing this practical, the student will be able to **simulate page replacement algorithms** and understand how different techniques like FIFO, LRU, and Optimal affect memory management and system performance. This outcome aligns with **CO4** — "Simulate Disk scheduling, Memory allocation, File allocation, and Page replacement algorithms."
Aim	Simulate the page replacement algorithm (Any Two)
Problem Definition	1. First In First Out (FIFO) 2. Least Recently Used (LRU) 3. Optimal
Theory (100 words)	Introduction: • Page replacement algorithms are used in operating systems to manage memory when a page fault occurs. • When all memory frames are full and a new page needs to be loaded, the system decides which existing page to replace. Objective: • To reduce the number of page faults and improve overall CPU performance. • Efficient page replacement ensures better utilization of physical memory. First In First Out (FIFO):

<table>
<tr><td></td><td>

This is the simplest page replacement technique.
Pages are replaced in the order they were loaded into memory — the oldest page is replaced first.
It uses a queue to track the order of pages.
Although easy to implement, it may lead to Belady's anomaly (more frames causing more faults).

□ Least Recently Used (LRU):

Replaces the page that has not been used for the longest time.
It assumes pages used recently are likely to be used again soon.
Implemented using counters or stacks to track recent usage.
It gives better performance than FIFO but is more complex.

□ Optimal Page Replacement:

Replaces the page that will not be used for the longest period in the future.
It gives the minimum possible page faults but is theoretical as future references are unknown.

</td></tr>
<tr><td>Procedure and Execution

(100 Words)</td><td>
Procedure:

Start the Program.
Initialize the number of frames and the reference string (page sequence).
Input Data.
Enter the total number of pages and their reference order.
Enter the number of available memory frames.
Select Algorithm.
Choose any two algorithms (e.g., FIFO and LRU or FIFO and Optimal) for simulation.
Simulate Page Replacement.
For each page reference:
Check if the page is already present in memory.
If yes, it's a hit (no replacement).
If no, it's a page fault and a replacement is done according to the chosen algorithm.
Count Page Faults.
Maintain a counter to record the number of page faults for each algorithm.
Display Results.
Show the step-by-step status of frames after each page reference.
Display the total number of page faults and hits for each

</td></tr>
</table>

algorithm.

7. End the Program.

Execution:

1. Input Example:
   - Reference String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3
   - Number of Frames: 3
2. For FIFO:
   - Pages are replaced in the order they arrive.
   - Total Page Faults: 9
3. For LRU:
   - Replace the page least recently used.
   - Total Page Faults: 7
4. Output:
   - Display a table showing frame contents after each reference.
   - Finally, print total page faults for both algorithms for comparison.

Code:

```
#include <stdio.h>
#include <stdbool.h> // For using bool type

#define MAX_FRAMES 3 // Maximum number of frames in memory

void fifo(int pages[], int num_pages) {
    int frames[MAX_FRAMES];
    // Initialize frames with -1 to indicate empty
    for (int i = 0; i < MAX_FRAMES; i++) {
        frames[i] = -1;
    }

    int page_faults = 0;
    int next_frame_index = 0; // Index to replace the oldest page

    printf("Page Reference String: ");
    for (int i = 0; i < num_pages; i++) {
        printf("%d ", pages[i]);
    }
    printf("\n\n");
```

	See code below

```
    for (int i = 0; i < num_pages; i++) {
        int current_page = pages[i];
        bool found = false;

        // Check if the page is already in a frame (page hit)
        for (int j = 0; j < MAX_FRAMES; j++) {
            if (frames[j] == current_page) {
                found = true;
                break;
            }
        }

        // If page not found (page fault)
        if (!found) {
            frames[next_frame_index] = current_page; // Place new page
            next_frame_index = (next_frame_index + 1) % MAX_FRAMES; //
Move to next frame for replacement
            page_faults++;
        }

        // Print current state of frames
        printf("Page %d -> Frames: [", current_page);
        for (int j = 0; j < MAX_FRAMES; j++) {
            if (frames[j] != -1) {
                printf(" %d ", frames[j]);
            } else {
                printf(" - ");
            }
        }
        printf("]\n");
    }

    printf("\nTotal Page Faults: %d\n", page_faults);
}

int main() {
    int pages[] = {1, 3, 0, 3, 5, 6, 2, 4, 3, 1, 0}; // Example page reference string
    int num_pages = sizeof(pages) / sizeof(pages[0]);

    fifo(pages, num_pages);

    return 0;
```

Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri, Nagpur - 441 110
NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu
Department of Computer Technology B. Tech in Computer Science and Engineering (IOT)

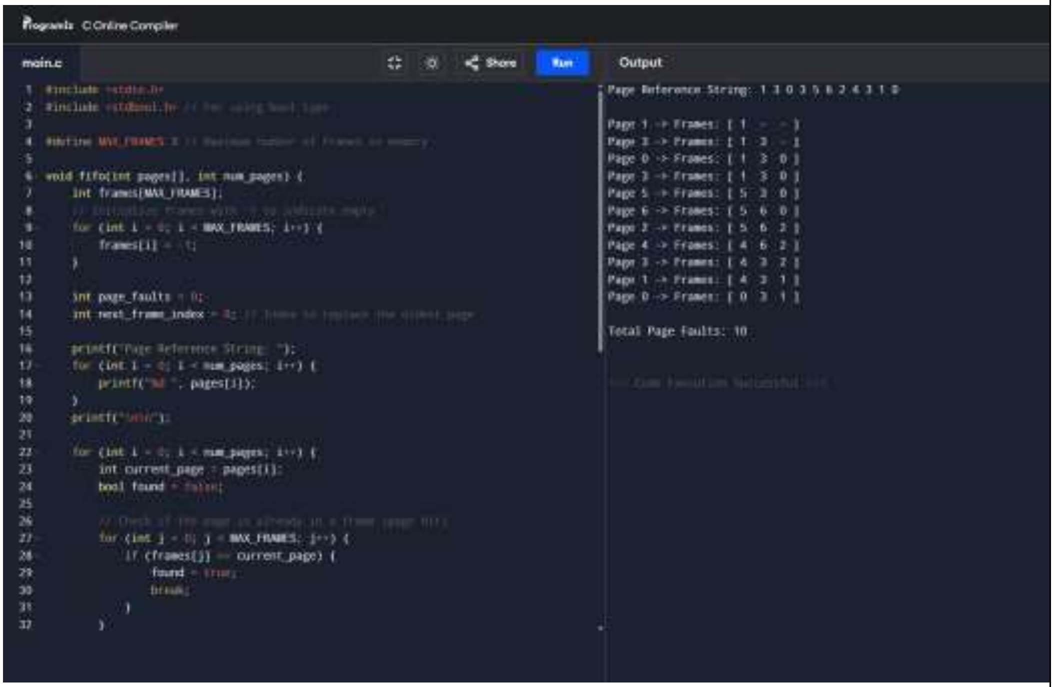
Vision of the Department

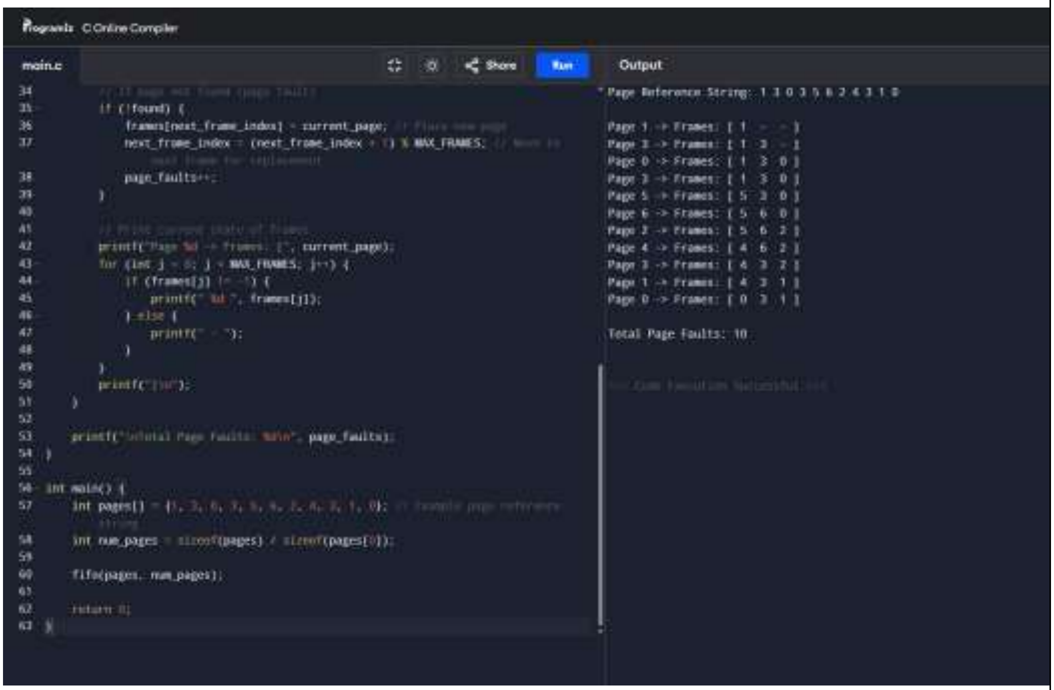
To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

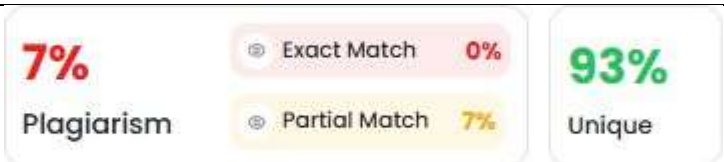
Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.

Output:





Output Analysis	The output of the page replacement simulation shows how different algorithms manage memory efficiency. Using the reference string (7, 0, 1, 2, 0, 3, 0, 4, 2, 3) with three frames, the FIFO algorithm resulted in 9 page faults, as it replaces pages in the order they arrived without considering recent usage. In contrast, the LRU algorithm produced only 7 page faults, since it replaces the page that hasn’t been used for the longest time. This analysis indicates that LRU performs better than FIFO by reducing page faults and improving memory utilization, making it a more efficient page replacement strategy.
Link of student Github profile where lab assignment has been uploaded	https://github.com/karan-0123/OS_LAB
Conclusion	The page replacement algorithm simulation concludes that efficient memory management greatly affects system performance. Among the algorithms tested, **LRU performs better than FIFO** by reducing page faults and improving memory utilization. Thus, choosing the right page replacement strategy helps in achieving faster and more efficient process execution in operating systems.
Plag Report (Similarity index < 12%)	
Date	25/10/2025