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

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: Dream of where you want.	**Mission:** Means to achieve Vision

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)

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

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	**2025-26 (ODD)**	**Course Name**	**Operating System Lab**
Semester	**5**	**Course Code**	**23IOT1504**
Roll No	**35**	**Name of Student**	**Bhushan Tayade**

Practical Number	**6**
Course Outcome	1. Understand Computer System Configuration and Simulate system resources efficiently using Linux Commands (CO1) 2. Analyse operating system functionalities utilizing system calls, thread programming and process scheduling algorithms (CO2) 3. Apply Synchronization primitives to implement a Deadlock-free solution(CO3) 4. Simulate Disk scheduling, Memory allocation, File allocation, page replacement algorithms (CO4)
Aim	Simulate the Page Replacement Algorithms: A. First In First Out(FIFO) B. Least Replacement Used(LRU) C. Optimal
Problem Definition	
Theory (100 words)	Page replacement algorithms manage memory by deciding which pages to replace when a page fault occurs. First In First Out (FIFO) replaces the oldest loaded page, following a simple queue-based approach but may cause Belady's anomaly. Least Recently Used (LRU) replaces the page that hasn't been used for the longest time, assuming past usage predicts future use, thus improving efficiency. Optimal Page Replacement replaces the page that will not be used for the longest time in the future, providing the best possible performance but requiring future knowledge. These algorithms aim to minimize page faults and improve system performance.
Procedure and Execution (100 Words)	Step for Implementation: A. Input the number of frames and the reference string. B. Initialize an empty list or queue to represent memory frames. C. For each page in the reference string: D. Check if the page is already in memory (page hit). E. If not (page fault), apply the replacement policy:

a. **FIFO:** Replace the oldest page in the queue.
b. **LRU:** Replace the least recently used page.
c. **Optimal:** Replace the page not needed for the longest future time.

F. Update frames after each operation.
G. Count and display page faults and final frame status.

Code: 1) First In First Out-

```
#include <stdio.h>

int main() {
   int pages[30], frames[10], n, f, i, j, k, pageFaults = 0, next = 0;
   printf("Enter number of pages: ");
   scanf("%d", &n);
   printf("Enter reference string: ");
   for (i = 0; i < n; i++)
      scanf("%d", &pages[i]);
   printf("Enter number of frames: ");
   scanf("%d", &f);

   for (i = 0; i < f; i++)
      frames[i] = -1;

   printf("\nPage Replacement Process (FIFO):\n");

   for (i = 0; i < n; i++) {
      int found = 0;
      for (j = 0; j < f; j++)
         if (frames[j] == pages[i]) {
            found = 1;
            break;
         }

      if (!found) {
         frames[next] = pages[i];
         next = (next + 1) % f;
         pageFaults++;
      }

      printf("Step %2d: ", i + 1);
      for (k = 0; k < f; k++) {
         if (frames[k] != -1)
            printf("%d ", frames[k]);
```

```
            else
                printf("- ");
        }
        printf("\n");
    }

    printf("\nTotal Page Faults (FIFO): %d\n", pageFaults);
    return 0;
}
```

2) Least Replacement Used(LRU)-

```
#include <stdio.h>

int findLRU(int time[], int n) {
    int min = time[0], pos = 0;
    for (int i = 1; i < n; i++)
        if (time[i] < min) {
            min = time[i];
            pos = i;
        }
    return pos;
}

int main() {
    int pages[30], frames[10], time[10], counter = 0;
    int n, f, i, j, pos, pageFaults = 0;
    printf("Enter number of pages: ");
    scanf("%d", &n);
    printf("Enter reference string: ");
    for (i = 0; i < n; i++)
        scanf("%d", &pages[i]);
    printf("Enter number of frames: ");
    scanf("%d", &f);

    for (i = 0; i < f; i++)
        frames[i] = -1;

    printf("\nPage Replacement Process (LRU):\n");

    for (i = 0; i < n; i++) {
        int found = 0;
        for (j = 0; j < f; j++)
            if (frames[j] == pages[i]) {
                found = 1;
```

```
            time[j] = ++counter;
            break;
         }

      if (!found) {
         int empty = -1;
         for (j = 0; j < f; j++)
            if (frames[j] == -1) {
               empty = j;
               break;
            }

         if (empty != -1)
            pos = empty;
         else
            pos = findLRU(time, f);

         frames[pos] = pages[i];
         time[pos] = ++counter;
         pageFaults++;
      }

      printf("Step %2d: ", i + 1);
      for (j = 0; j < f; j++) {
         if (frames[j] != -1)
            printf("%d ", frames[j]);
         else
            printf("- ");
      }
      printf("\n");
   }

   printf("\nTotal Page Faults (LRU): %d\n", pageFaults);
   return 0;
}

3) Optimal-
#include <stdio.h>

int predict(int pages[], int frames[], int n, int index, int f) {
   int farthest = index, res = -1;
   for (int i = 0; i < f; i++) {
      int j;
      for (j = index; j < n; j++) {
```

```
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    res = i;
                }
                break;
            }
        }
        if (j == n)
            return i;
    }
    return (res == -1) ? 0 : res;
}

int main() {
    int pages[30], frames[10], n, f, i, j, pos, pageFaults = 0;
    printf("Enter number of pages: ");
    scanf("%d", &n);
    printf("Enter reference string: ");
    for (i = 0; i < n; i++)
        scanf("%d", &pages[i]);
    printf("Enter number of frames: ");
    scanf("%d", &f);

    for (i = 0; i < f; i++)
        frames[i] = -1;

    printf("\nPage Replacement Process (Optimal):\n");

    for (i = 0; i < n; i++) {
        int found = 0;
        for (j = 0; j < f; j++)
            if (frames[j] == pages[i]) {
                found = 1;
                break;
            }

        if (!found) {
            int empty = -1;
            for (j = 0; j < f; j++)
                if (frames[j] == -1) {
                    empty = j;
                    break;
                }
```

```
            if (empty != -1)
                pos = empty;
            else
                pos = predict(pages, frames, n, i + 1, f);

            frames[pos] = pages[i];
            pageFaults++;
        }

        printf("Step %2d: ", i + 1);
        for (j = 0; j < f; j++) {
            if (frames[j] != -1)
                printf("%d ", frames[j]);
            else
                printf("- ");
        }
        printf("\n");
    }

    printf("\nTotal Page Faults (Optimal): %d\n", pageFaults);
    return 0;
}
```

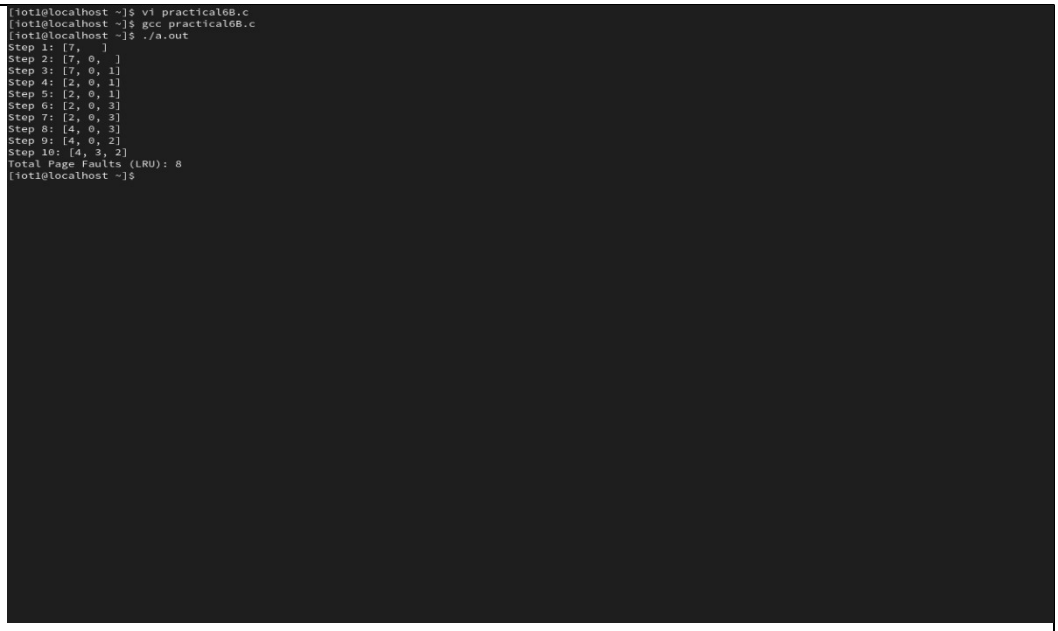
Output:

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.

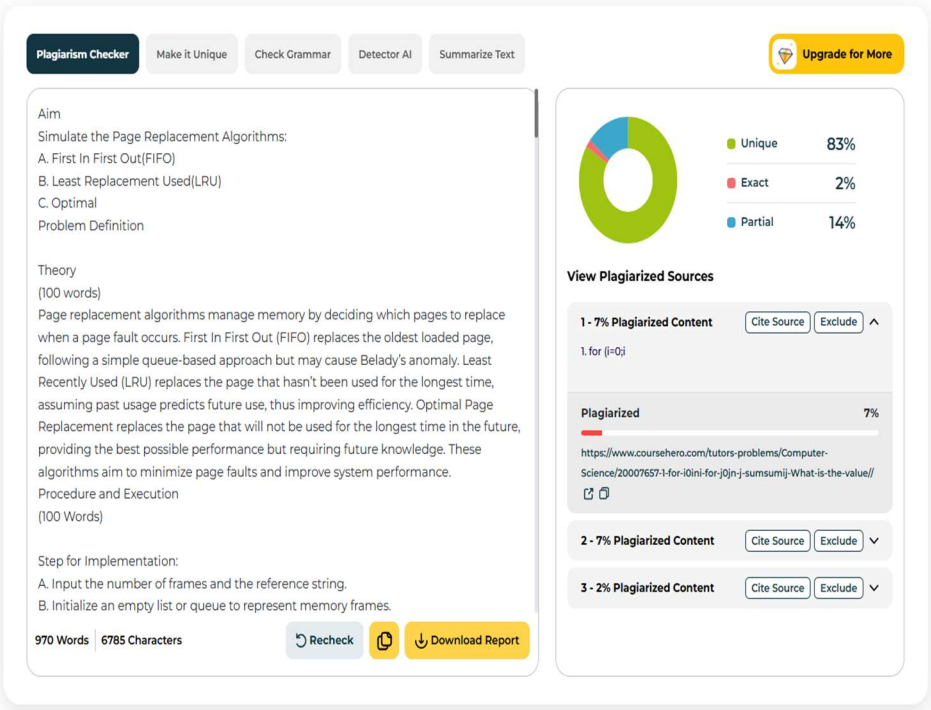
Output Analysis	 
Link of student Github profile where lab	"https://github.com/Bhushan-Tayade/YCCN-23071391.git"

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.

assignment has been uploaded	
Conclusion	In this practical, we simulated and analyzed three major page replacement algorithms — **FIFO**, **LRU**, and **Optimal**. Each algorithm manages memory pages differently when a page fault occurs. The **FIFO (First In First Out)** algorithm is simple but may lead to more page faults due to Belady's anomaly. The **LRU (Least Recently Used)** algorithm performs better by replacing the least recently used page, based on past usage. The **Optimal** algorithm gives the best possible performance by replacing the page that will not be used for the longest period, though it is impractical for real systems as it requires future knowledge.
Plag Report (Similarity index < 12%)	**Result** 
Date	**30/10/2025**