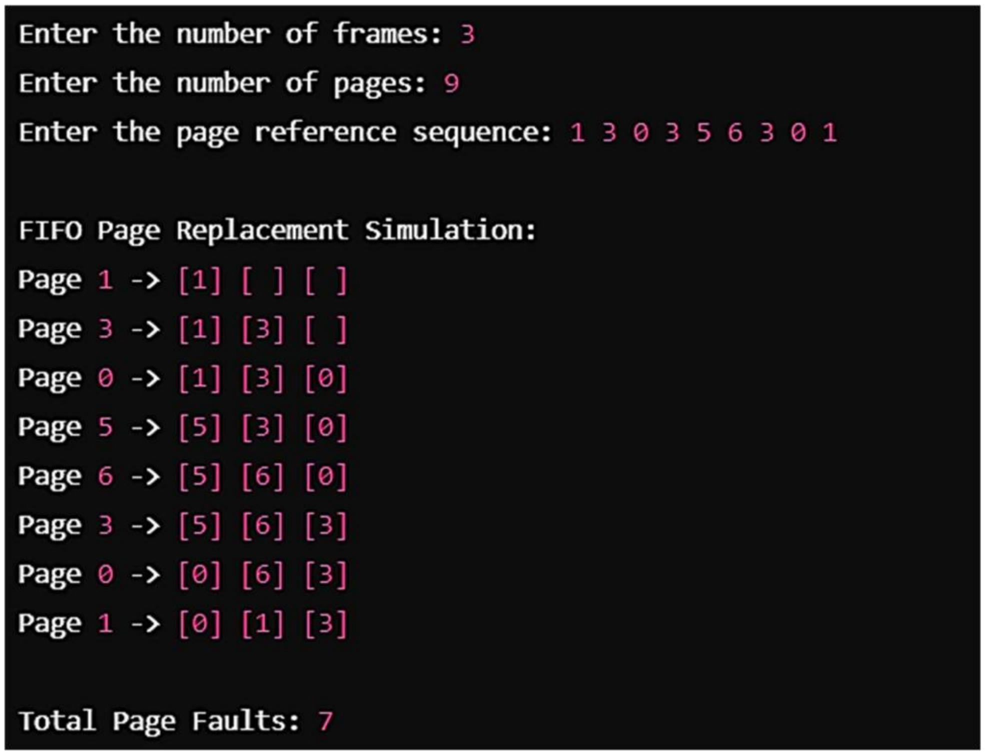
PROJECT - 1

AIM: Write a program to simulate First In First Out (FIFO), Least Recently Used (LRU) page replacement algorithms.

1. **FIFO Page Replacement (fifo.c)**

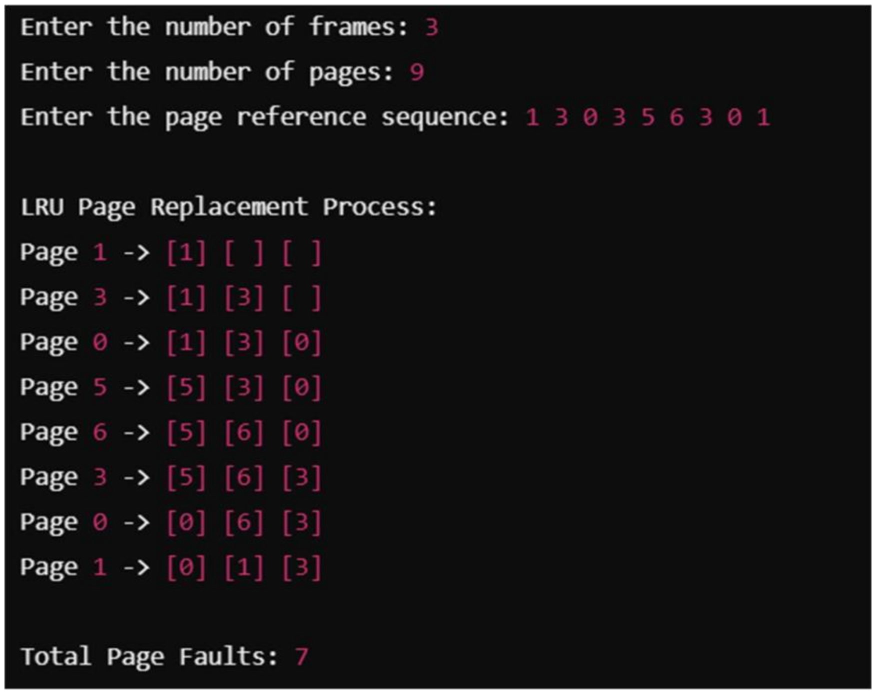
```
#include <stdio.h>
int main() {
    int frames, pages, i, j, k, pageFaults = 0, index = 0;
    int frameQueue[10], pageSequence[30];
    printf("Enter the number of frames: ");
    scanf("%d", &frames);
    printf("Enter the number of pages: ");
    scanf("%d", &pages);
    printf("Enter the page reference sequence: ");
    for (i = 0; i < pages; i++) {
        scanf("%d", &pageSequence[i]);
    }
    for (i = 0; i < frames; i++) {
        frameQueue[i] = -1;
    }
    printf("\nFIFO Page Replacement Simulation:\n");

    for (i = 0; i < pages; i++) {
        int page = pageSequence[i];
        int found = 0;

        for (j = 0; j < frames; j++) {
            if (frameQueue[j] == page) {
                found = 1;
                break;
            }
        }

        if (!found) {
            frameQueue[index] = page;
            index = (index + 1) % frames;
            pageFaults++;

            printf("Page %d -> ", page);
            for (k = 0; k < frames; k++) {
               if (frameQueue[k] == -1)
                    printf("[ ] ");
                else
                    printf("[%d] ", frameQueue[k]);
            }
            printf("\n");
        }
    }

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



```
Enter the number of frames: 3
Enter the number of pages: 9
Enter the page reference sequence: 1 3 0 3 5 6 3 0 1

FIFO Page Replacement Simulation:
Page 1 -> [1] [ ] [ ]
Page 3 -> [1] [3] [ ]
Page 0 -> [1] [3] [0]
Page 5 -> [5] [3] [0]
Page 6 -> [5] [6] [0]
Page 3 -> [5] [6] [3]
Page 0 -> [0] [6] [3]
Page 1 -> [0] [1] [3]

Total Page Faults: 7
```

2. LRU Page Replacement Algorithm

```
#include <stdio.h>
int main() {
    int frames, pages;
    int pageFaults = 0;
    int frameQueue[10], usedRecently[10];
    int pageSequence[30];
    printf("Enter the number of frames: ");
    scanf("%d", &frames);
    printf("Enter the number of pages: ");
    scanf("%d", &pages);
    printf("Enter the page reference sequence: ");
    for (int i = 0; i < pages; i++) {
        scanf("%d", &pageSequence[i]);
    }
    for (int i = 0; i < frames; i++) {
        frameQueue[i] = -1;
    }
    printf("\nLRU Page Replacement Process:\n");
    for (int i = 0; i < pages; i++) {
        int page = pageSequence[i];
        int found = 0;
        for (int j = 0; j < frames; j++) {
            if (frameQueue[j] == page) {
                found = 1;
                usedRecently[j] = i;
                break;
            }
        }
        if (!found) {
            int replaceIndex = 0;
            for (int j = 0; j < frames; j++) {
                if (frameQueue[j] == -1) {
                    replaceIndex = j;
                    break;
                }
            }
            if (frameQueue[replaceIndex] != -1) {
                int lru = usedRecently[0];
                for (int j = 1; j < frames; j++) {
                    if (usedRecently[j] < lru) {
                        lru = usedRecently[j];
                        replaceIndex = j;
                    }
                }
            }
            frameQueue[replaceIndex] = page;
            usedRecently[replaceIndex] = i;
            pageFaults++;
            printf("Page %d -> ", page);
            for (int k = 0; k < frames; k++) {
                if (frameQueue[k] == -1)
                    printf("[ ] ");
                else
                    printf("[%d] ", frameQueue[k]);
            }
            printf("\n");
        }
    }
    printf("\nTotal Page Faults: %d\n", pageFaults);
    return 0;
}
```

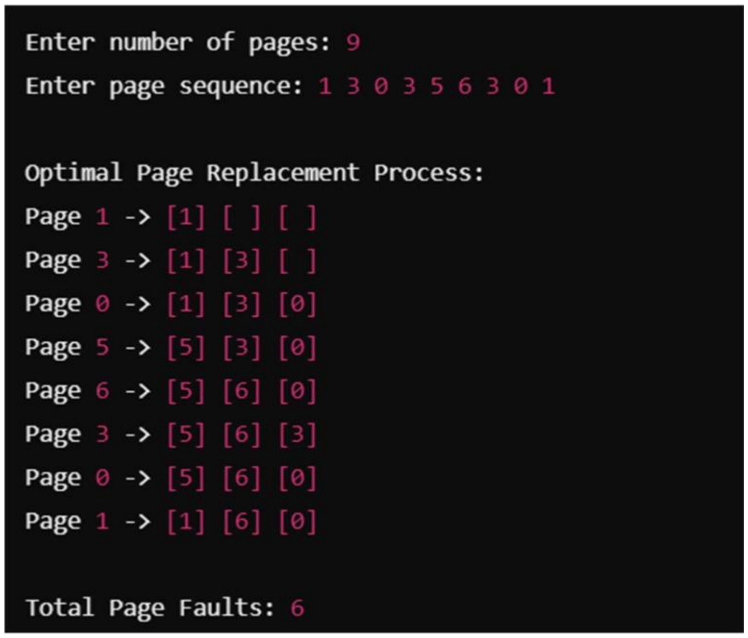


```
Enter the number of frames: 3
Enter the number of pages: 9
Enter the page reference sequence: 1 3 0 3 5 6 3 0 1

LRU Page Replacement Process:
Page 1 -> [1] [ ] [ ]
Page 3 -> [1] [3] [ ]
Page 0 -> [1] [3] [0]
Page 5 -> [5] [3] [0]
Page 6 -> [5] [6] [0]
Page 3 -> [5] [6] [3]
Page 0 -> [0] [6] [3]
Page 1 -> [0] [1] [3]

Total Page Faults: 7
```

3. Optimal Page Replacement Algorithm

```
#include <stdio.h>


int findOptimal(int pages[], int frames[], int n, int index, int f) {
    int farthest = -1, pos = -1;
    for (int i = 0; i < f; i++) {
        int j;
        for (j = index + 1; j < n; j++) {
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    pos = i;
                }
                break;
            }
        }
        if (j == n) return i;
    }
    return (pos == -1) ? 0 : pos;
```

```
}

int main() {
    int frames[3], pages[20], f = 3, n, pageFaults = 0;

    printf("Enter number of pages: ");
    scanf("%d", &n);
    printf("Enter page sequence: ");
    for (int i = 0; i < n; i++)
        scanf("%d",  &pages[i]);

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

    printf("\nOptimal  Page  Replacement  Process:\n");

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

        if (!found) {
            int replaceIndex = (i < f) ? i : findOptimal(pages, frames, n, i, f);
            frames[replaceIndex] = pages[i];
            pageFaults++;

            printf("Page %d -> ", pages[i]);
            for (int j = 0; j < f; j++)
                printf("[%d] ", frames[j]);
            printf("\n");
        }
    }

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



```
Enter number of pages: 9
Enter page sequence: 1 3 0 3 5 6 3 0 1

Optimal Page Replacement Process:
Page 1 -> [1] [ ] [ ]
Page 3 -> [1] [3] [ ]
Page 0 -> [1] [3] [0]
Page 5 -> [5] [3] [0]
Page 6 -> [5] [6] [0]
Page 3 -> [5] [6] [3]
Page 0 -> [5] [6] [0]
Page 1 -> [1] [6] [0]

Total Page Faults: 6
```