A Harsha Mukundha - 192324070

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

AIM

To construct a C program to simulate the **First-In-First-Out (FIFO)** paging technique of memory management, which replaces the oldest page in memory when a new page needs to be loaded and all frames are full.

ALGORITHM

1. **Start**
2. Input the total number of pages, the sequence of page references, and the number of available frames.
3. Initialize the frames as empty (-1) and set the page fault counter to 0.
4. For each page in the reference sequence:
    - Check if the page is already present in any of the frames.
    - If the page is found in the frames, move to the next page (no page fault).
    - If the page is not found:
        - Replace the oldest page in the frames using the FIFO approach.
        - Increment the page fault counter.
    - Update the frame contents and display the current frame status.
5. Display the total number of page faults after processing all pages.
6. **Stop**

PROCEDURE

1. Include necessary libraries.
2. Define variables for frame size, pages, page faults, and an array to represent frames.
3. Take input for the number of pages, the sequence of page references, and the number of frames.
4. Use a loop to process each page reference in the sequence:

- Check if the page is already in a frame.
- If not, replace the oldest page in the frame using the FIFO technique.
- Increment the page fault counter.

5. Display the frame status after each page reference.
6. Print the total number of page faults.

CODE:

```
#include <stdio.h>

void fifoPaging(int pages[], int n, int frames[], int f) {
  int pageFaults = 0, index = 0, found, i, j;

  printf("Page Reference\tFrames\n");
  for (i = 0; i < n; i++) {
    found = 0;

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

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

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

    printf("Total Page Faults: %d\n", pageFaults);
}

int main() {
    int n, f, i;

    printf("Enter the number of pages: ");
    scanf("%d", Cn);

    int pages[n];
    printf("Enter the page reference sequence: ");
    for (i = 0; i < n; i++) {
        scanf("%d", Cpages[i]);
    }

    printf("Enter the number of frames: ");
    scanf("%d", Cf);
```

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

    fifoPaging(pages, n, frames, f);

    return 0;
}
```

OUTPUT: