/*fifo page replacement*/

```
#include <stdio.h>


void printFrames(int frames[], int n) {
   for (int i = 0; i < n; i++) {
      // If the frame is empty, print '-'
      if (frames[i] == -1) {
         printf(" - ");
      } else {
         // Otherwise, print the page number in the frame
         printf(" %d ", frames[i]);
      }
   }
   printf("\n");
}

int isPagePresent(int frames[], int n, int page) {
   for (int i = 0; i < n; i++) {
      // If the page is found in the frames, return 1 (true)
      if (frames[i] == page) {
         return 1;
      }
   }
   // If the page is not found, return 0 (false)[do nothing]
   return 0;
}



void FIFO(int pages[], int n, int maxFrames) {
   int frames[maxFrames];
   // Index to keep track of the front of the frames (for replacement)
```

```
int front = 0;
// Variable to keep track of the number of page faults
int pageFaults = 0;

// Initialize frames with -1 indicating an empty frame
for (int i = 0; i < maxFrames; i++) {
   frames[i] = -1;
}

printf("Page\tFrames\tPage Faults\n");

for (int i = 0; i < n; i++) {
   // Print the current page reference
  printf("%d\t", pages[i]);

   // Check if the page is present in the frames
   if (!isPagePresent(frames, maxFrames, pages[i])) {
      // If not present, it's a page fault
      pageFaults++;
      // Replace the page at the front of the frames with the new page
      frames[front] = pages[i];
      front = (front + 1) % maxFrames;
   }

   // Print the current state of frames
   printFrames(frames, maxFrames);
   // Print the current number of page faults
   printf("\t\t\t%d\n", pageFaults);
}

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

```
}


int main() {
    int n;
    printf("Enter the number of page references: ");
    scanf("%d", &n);
    int maxFrames;
    printf("Enter the maximum number of frames: ");
    scanf("%d", &maxFrames);
    int pages[n];
    printf("Enter the page reference sequence:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &pages[i]);
    }
    FIFO(pages, n, maxFrames);
    return 0;
}
```