Ex No.: 11c OPTIMAL PAGE REPLACEMENT

Date : 23.04.2025

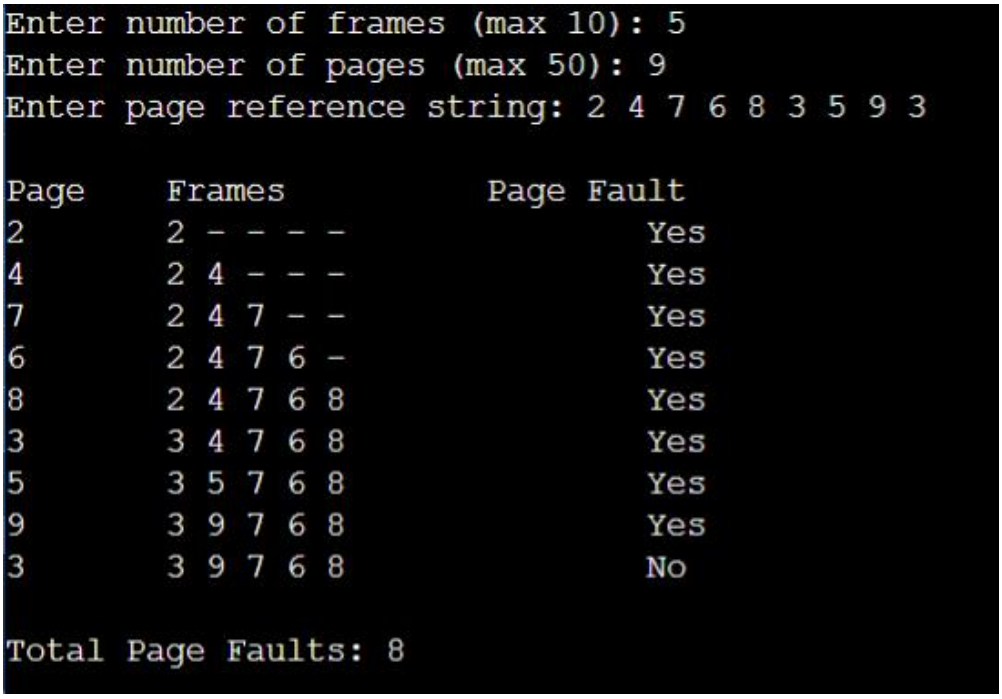
Aim :

To write a c program to implement Optimal page replacement algorithm.

Code:

```
#include <stdio.h>
#include <stdbool.h>

#define MAX_FRAMES 10
#define MAX_PAGES 50

int findOptimal(int pages[], int frames[], int start, int end, int n) {
    int farthest = start;
    int idx = -1;

    for (int i = 0; i < n; i++) {
        int j;
        for (j = start; j <= end; j++) {
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    idx = i;
                }
                break;
            }
        }
        if (j == end + 1) {
            return i;
        }
    }
    if (idx == -1) {
        return 0;
    }
    else {
    return idx;
    }
}

int main() {
    int frames[MAX_FRAMES], pages[MAX_PAGES];
    int n_frames, n_pages, page_faults = 0;
    printf("Enter number of frames (max %d): ", MAX_FRAMES);
    scanf("%d", &n_frames);
    printf("Enter number of pages (max %d): ", MAX_PAGES);
    scanf("%d", &n_pages);
    printf("Enter page reference string: ");
    for (int i = 0; i < n_pages; i++) {
        scanf("%d", &pages[i]);
    }

    for (int i = 0; i < n_frames; i++) {
        frames[i] = -1;
    }
    printf("\nPage\tFrames\t\tPage Fault\n");
    for (int i = 0; i < n_pages; i++) {
        bool page_found = false;

        for (int j = 0; j < n_frames; j++) {
            if (frames[j] == pages[i]) {
                page_found = true;
                break;
            }
        }
        if (!page_found) {
            int pos;
            bool has_empty = false;
            for (int j = 0; j < n_frames; j++) {
                if (frames[j] == -1) {
                    pos = j;
                    has_empty = true;
                    break;
                }
            }
            if (!has_empty) {
                pos = findOptimal(pages, frames, i + 1, n_pages - 1, n_frames);
            }

            frames[pos] = pages[i];
            page_faults++;
        }
        printf("%d\t", pages[i]);
        for (int j = 0; j < n_frames; j++) {
            if (frames[j] != -1)
                printf("%d ", frames[j]);
            else
                printf("- ");
        }
        printf("\t\t%s\n", page_found ? "No" : "Yes");
    }
    printf("\nTotal Page Faults: %d\n", page_faults);
    return 0;
}
```

Output:



```
Enter number of frames (max 10): 5
Enter number of pages (max 50): 9
Enter page reference string: 2 4 7 6 8 3 5 9 3

Page    Frames          Page Fault
2       2 - - - -               Yes
4       2 4 - - -               Yes
7       2 4 7 - -               Yes
6       2 4 7 6 -               Yes
8       2 4 7 6 8               Yes
3       3 4 7 6 8               Yes
5       3 5 7 6 8               Yes
9       3 9 7 6 8               Yes
3       3 9 7 6 8               No

Total Page Faults: 8
```

Result:

Thus the program to find out the number of page faults that occur using Optimal page replacement technique has been executed successfully.