OPT

```c
#include <stdio.h>

void display(int frames[], int n) {
    for (int i = 0; i < n; i++) {
        printf(frames[i] == -1 ? " - " : " %d ", frames[i]);
    }
    printf("\n");
}

int isPageInFrame(int page, int frames[], int n) {
    for (int i = 0; i < n; i++) {
        if (frames[i] == page) return 1;
    }
    return 0;
}

int findOptimal(int frames[], int n, int ref_string[], int ref_len, int current_index) {
    int farthest = -1, index = -1;
    for (int i = 0; i < n; i++) {
        int j;
        for (j = current_index; j < ref_len; j++) {
            if (frames[i] == ref_string[j]) {
                if (j > farthest) {
                    farthest = j;
                    index = i;
                }
                break;
            }
        }
        if (j == ref_len) return i;
    }
    return (index != -1) ? index : 0;
}

void runOptimal(int ref_string[], int ref_len, int n) {
    int frames[n], page_faults = 0;
    for (int i = 0; i < n; i++) frames[i] = -1;

    printf("Page Replacement Process (Optimal):\n");
    for (int i = 0; i < ref_len; i++) {
        int page = ref_string[i];
        if (!isPageInFrame(page, frames, n)) {
            int replace_index = findOptimal(frames, n, ref_string, ref_len, i);
            frames[replace_index] = page;
            page_faults++;
        }
        display(frames, n);
    }
    printf("Total Page Faults: %d\n\n", page_faults);
}

int main() {
    int n;
    printf("Enter number of frames: ");
    scanf("%d", &n);

    int ref_string[] = {3, 4, 5, 6, 3, 4, 7, 3, 4, 5, 6, 7, 2, 4, 6};
    int ref_len = sizeof(ref_string) / sizeof(ref_string[0]);

    runOptimal(ref_string, ref_len, n);
    return 0;
}
```