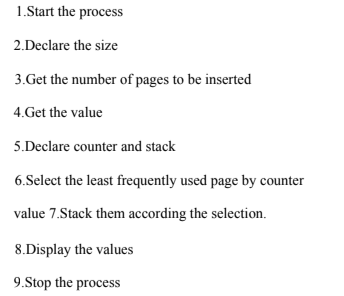
| **Ex no:11 c)** | **OPTIMAL** |
| --- | --- |

**Aim:**

****

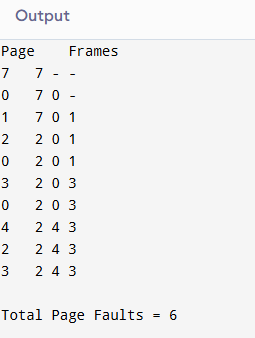
**Algorithm:**

****

**Program:**

#include <stdio.h>  
  
int main() {  
    int pages[] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3};  // Reference String  
    int n = 10;                                    // Number of pages  
    int frames = 3;                                // Number of frames  
    int frame[10];  
    int pageFaults = 0;  
  
    // Initialize all frames as empty (-1)  
    for (int i = 0; i < frames; i++) {  
        frame[i] = -1;  
    }  
  
    printf("Page\tFrames\n");  
  
    for (int i = 0; i < n; i++) {  
        int found = 0;  
  
        // Check if page is already in frame  
        for (int j = 0; j < frames; j++) {  
            if (frame[j] == pages[i]) {  
                found = 1;  
                break;  
            }  
        }  
  
        // If page not found - Page Fault  
        if (found == 0) {  
            pageFaults++;  
  
            // Check for empty frame  
            int empty = -1;  
            for (int j = 0; j < frames; j++) {  
                if (frame[j] == -1) {  
                    empty = j;  
                    break;  
                }  
            }  
  
            // If empty frame available  
            if (empty != -1) {  
                frame[empty] = pages[i];  
            } else {  
                int farthest = -1, pos = -1;  
  
                // Find the page not used for longest time  
                for (int j = 0; j < frames; j++) {  
                    int k;  
                    for (k = i + 1; k < n; k++) {  
                        if (frame[j] == pages[k])  
                            break;  
                    }  
  
                    if (k > farthest) {  
                        farthest = k;  
                        pos = j;  
                    }  
                }  
  
                frame[pos] = pages[i];  // Replace page  
            }  
        }  
  
        // Display current frame status  
        printf("%d\t", pages[i]);  
        for (int j = 0; j < frames; j++) {  
            if (frame[j] != -1)  
                printf("%d ", frame[j]);  
            else  
                printf("- ");  
        }  
        printf("\n");  
    }  
  
    printf("\nTotal Page Faults = %d\n", pageFaults);  
  
    return 0;  
}

**Output:**

****

**Result:**

Thus the optimal page replacement program was executed successfully.