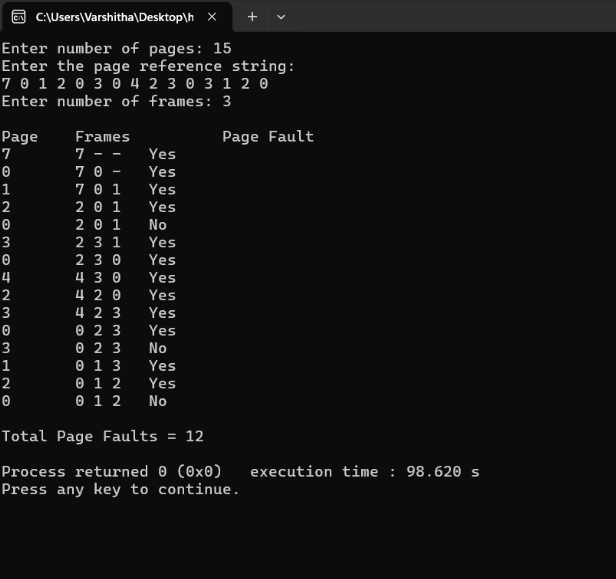
**Write a C program to simulate page replacement algorithms**

**a)FIFO**

#include <stdio.h>  
  
int main() {  
    int frames, pages[50], n, frame[10], time[10];  
    int i, j, k, avail, count = 0, current\_time = 0, pos, lru\_time;  
  
    printf("Enter number of pages: ");  
    scanf("%d", &n);  
  
    printf("Enter the page reference string:\n");  
    for(i = 0; i < n; i++)  
        scanf("%d", &pages[i]);  
  
    printf("Enter number of frames: ");  
    scanf("%d", &frames);  
  
    for(i = 0; i < frames; i++) {  
        frame[i] = -1;  
        time[i] = -1;  
    }  
  
    printf("\nPage\tFrames\t\tPage Fault\n");  
  
    for(i = 0; i < n; i++) {  
        avail = 0;  
        current\_time++;  
  
        for(j = 0; j < frames; j++) {  
            if(frame[j] == pages[i]) {  
                avail = 1;  
                time[j] = current\_time;  
                break;  
            }  
        }  
  
        if(avail == 0) {  
            pos = 0;  
            lru\_time = time[0];  
  
            for(j = 1; j < frames; j++) {  
                if(time[j] < lru\_time) {  
                    lru\_time = time[j];  
                    pos = j;  
                }  
            }  
  
            frame[pos] = pages[i];  
            time[pos] = current\_time;  
            count++;  
  
            printf("%d\t", pages[i]);  
            for(k = 0; k < frames; k++) {  
                if(frame[k] != -1)  
                    printf("%d ", frame[k]);  
                else  
                    printf("- ");  
            }  
            printf("\tYes\n");  
        } else {  
            printf("%d\t", pages[i]);  
            for(k = 0; k < frames; k++) {  
                if(frame[k] != -1)  
                    printf("%d ", frame[k]);  
                else  
                    printf("- ");  
            }  
            printf("\tNo\n");  
        }  
    }  
  
    printf("\nTotal Page Faults = %d\n", count);  
    return 0;  
}

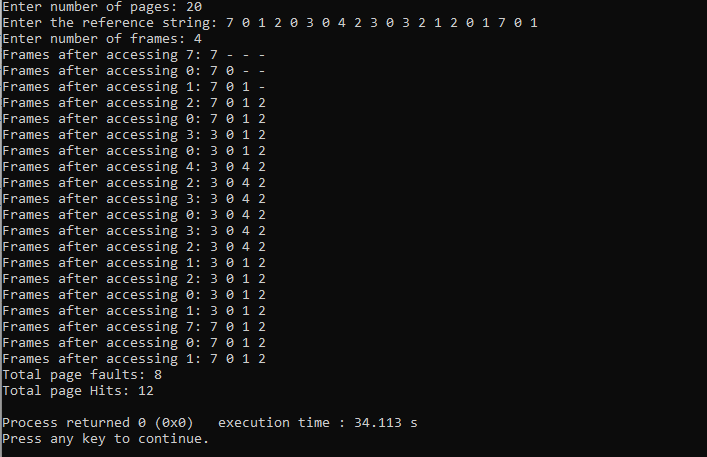
Output:

****

1. **LRU**

#include <stdio.h>  
  
int main() {  
    int n, frames, i, j, k, faults = 0;  
    printf("Enter number of pages: ");  
    scanf("%d", &n);  
    int pages[n];  
    printf("Enter the reference string: ");  
    for(i = 0; i < n; i++)  
        scanf("%d", &pages[i]);  
  
    printf("Enter number of frames: ");  
    scanf("%d", &frames);  
  
    int frame\_arr[frames];  
    int time[frames]; // To track the usage time  
    for(i = 0; i < frames; i++) {  
        frame\_arr[i] = -1;  
        time[i] = 0;  
    }  
  
    int counter = 0;  
    for(i = 0; i < n; i++) {  
        int flag = 0;  
        for(j = 0; j < frames; j++) {  
            if(frame\_arr[j] == pages[i]) {  
                flag = 1;  
                counter++;  
                time[j] = counter; // Update the usage time  
                break;  
            }  
        }  
  
        if(flag == 0) { // Page fault  
            faults++;  
            int min\_time = time[0], min\_pos = 0;  
            for(k = 1; k < frames; k++) {  
                if(time[k] < min\_time) {  
                    min\_time = time[k];  
                    min\_pos = k;  
                }  
            }  
            frame\_arr[min\_pos] = pages[i]; // Replace the least recently used  
            counter++;  
            time[min\_pos] = counter; // Update the usage time for the new page  
        }  
  
        printf("Frames after accessing %d: ", pages[i]);  
        for(j = 0; j < frames; j++) {  
            if(frame\_arr[j] == -1)  
                printf("- ");  
            else  
                printf("%d ", frame\_arr[j]);  
        }  
        printf("\n");  
    }  
  
    printf("Total page faults: %d\n", faults);  
    int Hits = n-faults;  
    printf("Total page Hits: %d\n",Hits);  
    return 0;  
}

**Output:**

****

**c) Optimal**

#include <stdio.h>

int main() {

    int n, frames, i, j, k, faults = 0;

    printf("Enter number of pages: ");

    scanf("%d", &n);

    int pages[n];

    printf("Enter the reference string: ");

    for(i = 0; i < n; i++)

        scanf("%d", &pages[i]);

    printf("Enter number of frames: ");

    scanf("%d", &frames);

    int frame\_arr[frames];

    for(i = 0; i < frames; i++)

        frame\_arr[i] = -1;

    for(i = 0; i < n; i++) {

        int flag = 0;

        for(j = 0; j < frames; j++) {

            if(frame\_arr[j] == pages[i]) {

                flag = 1;

                break;

            }

        }

        if(flag == 0) {

            faults++;

            int pos = -1;

            for(j = 0; j < frames; j++) {

                if(frame\_arr[j] == -1) {

                    pos = j;

                    break;

                }

            }

            if(pos == -1) {

                int farthest = i, replace\_index = 0;

                for(j = 0; j < frames; j++) {

                    int found = 0;

                    for(k = i + 1; k < n; k++) {

                        if(frame\_arr[j] == pages[k]) {

                            if(k > farthest) {

                                farthest = k;

                                replace\_index = j;

                            }

                            found = 1;

                            break;

                        }

                    }

                    if(!found) {

                        replace\_index = j;

                        break;

                    }

                }

                pos = replace\_index;

            }

            frame\_arr[pos] = pages[i];

        }

        printf("%d: ", pages[i]);

        for(j = 0; j < frames; j++) {

            if(frame\_arr[j] == -1)

                printf("\_ ");

            else

                printf("%d ", frame\_arr[j]);

        }

        printf("\n");

    }

    printf("Total page faults: %d\n", faults);

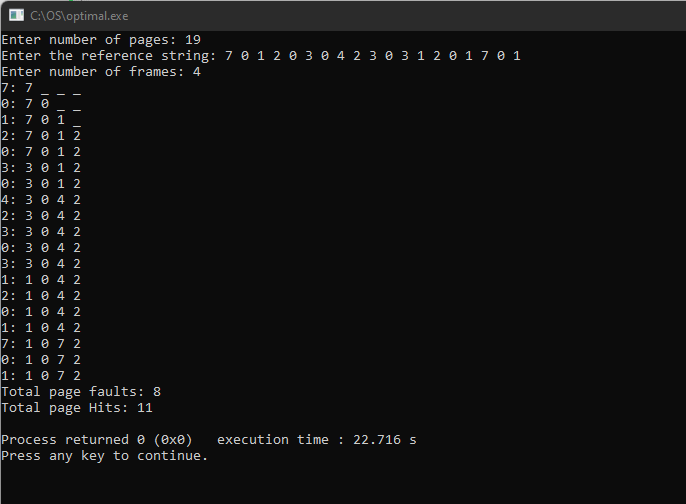
    int Hits = n-faults;

    printf("Total page Hits: %d\n",Hits);

    return 0;

}

Output:

****