**Name: Atharva Salitri Division: CSAI B**

**Roll No.: 37 PRN: 12310120**

**Subject: OS Lab Assignment 8**

**Title: Implementation of Page Replacement Strategies**

**1) First in First Out**

**Code:**

#include <stdio.h>

int main() {

int i, j, n, a[50], frame[10], no, k, avail, count = 0;

printf("\nENTER THE NUMBER OF PAGES: ");

scanf("%d", &n);

printf("\nENTER THE PAGE NUMBER: ");

for (i = 0; i < n; i++)

scanf("%d", &a[i]);

printf("\nENTER THE NUMBER OF FRAMES: ");

scanf("%d", &no);

for (i = 0; i < no; i++)

frame[i] = -1;

j = 0;

printf("Ref String\t Page Frames\n");

for (i = 0; i < n; i++) {

printf("%d\t\t", a[i]);

avail = 0;

for (k = 0; k < no; k++) {

if (frame[k] == a[i]) {

avail = 1;

break;

}

}

if (avail == 0) {

frame[j] = a[i];

j = (j + 1) % no;

count++;

}

for (k = 0; k < no; k++)

printf("%d\t", frame[k]);

printf("\n");

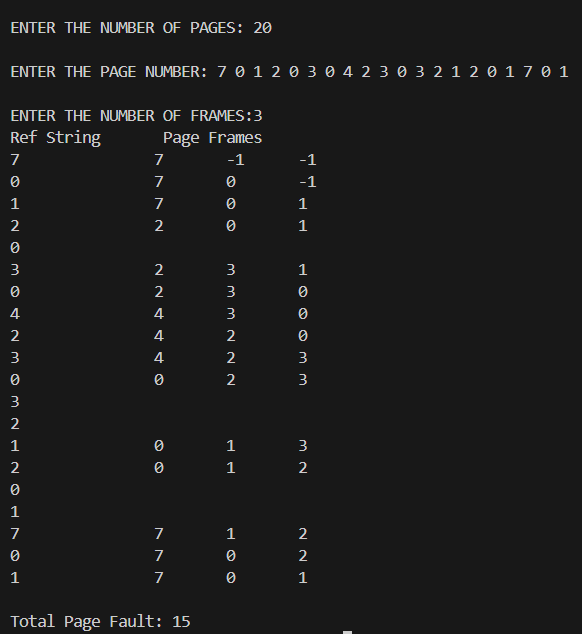
}

printf("\nTotal Page Fault: %d", count);

return 0;

}

**Output:**



**2) Optimal**

**Code:**

#include <stdio.h>

int main() {

int i, j, n, a[50], frame[10], no, k, count = 0, pos, flag1, flag2;

printf("\nENTER THE NUMBER OF PAGES: ");

scanf("%d", &n);

printf("\nENTER THE PAGE NUMBER: ");

for (i = 0; i < n; i++)

scanf("%d", &a[i]);

printf("\nENTER THE NUMBER OF FRAMES: ");

scanf("%d", &no);

for (i = 0; i < no; i++)

frame[i] = -1;

printf("Ref String\t Page Frames\n");

for (i = 0; i < n; i++) {

printf("%d\t\t", a[i]);

flag1 = flag2 = 0;

for (k = 0; k < no; k++) {

if (frame[k] == a[i]) {

flag1 = 1;

break;

}

}

if (flag1 == 0) {

for (k = 0; k < no; k++) {

if (frame[k] == -1) {

frame[k] = a[i];

flag2 = 1;

break;

}

}

if (flag2 == 0) {

pos = -1;

int farthest = -1;

for (k = 0; k < no; k++) {

int j;

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

if (frame[k] == a[j]) {

if (j > farthest) {

farthest = j;

pos = k;

}

break;

}

}

if (j == n) {

pos = k;

break;

}

}

frame[pos] = a[i];

}

count++;

for (k = 0; k < no; k++)

printf("%d\t", frame[k]);

} else {

for (k = 0; k < no; k++)

printf("\t");

}

printf("\n");

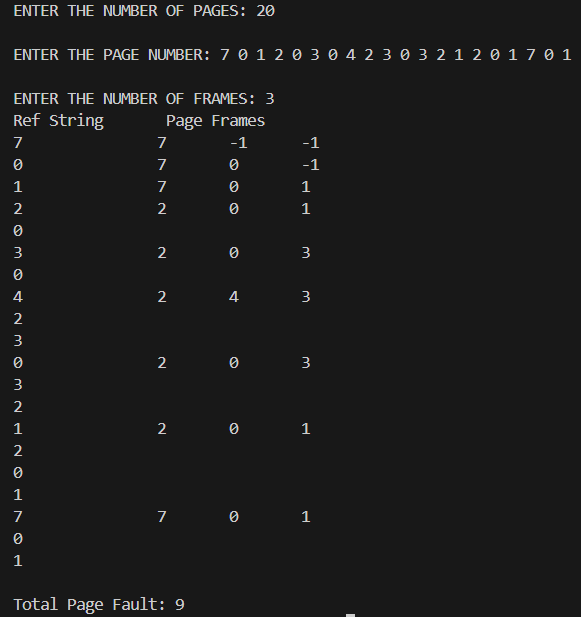
}

printf("\nTotal Page Fault: %d", count);

return 0;

}

**Output:**



**3) LRU**

**Code:**

#include <stdio.h>

int main() {

int i, j, n, a[50], frame[10], no, k, count = 0, pos, flag1;

int time[10];

printf("\nENTER THE NUMBER OF PAGES: ");

scanf("%d", &n);

printf("\nENTER THE PAGE NUMBER: ");

for (i = 0; i < n; i++)

scanf("%d", &a[i]);

printf("\nENTER THE NUMBER OF FRAMES: ");

scanf("%d", &no);

for (i = 0; i < no; i++)

frame[i] = -1;

for (i = 0; i < no; i++)

time[i] = 0;

printf("Ref String\t Page Frames\n");

for (i = 0; i < n; i++) {

printf("%d\t\t", a[i]);

flag1 = 0;

for (k = 0; k < no; k++) {

if (frame[k] == a[i]) {

flag1 = 1;

time[k] = 0;

break;

}

}

if (flag1 == 0) {

for (k = 0; k < no; k++) {

if (frame[k] == -1) {

frame[k] = a[i];

time[k] = 0;

flag1 = 1;

break;

}

}

if (flag1 == 0) {

pos = 0;

for (k = 1; k < no; k++) {

if (time[k] > time[pos]) {

pos = k;

}

}

frame[pos] = a[i];

time[pos] = 0;

}

count++;

for (k = 0; k < no; k++)

printf("%d\t", frame[k]);

} else {

for (k = 0; k < no; k++)

printf("\t");

}

for (k = 0; k < no; k++) {

if (frame[k] != -1) {

time[k]++;

}

}

printf("\n");

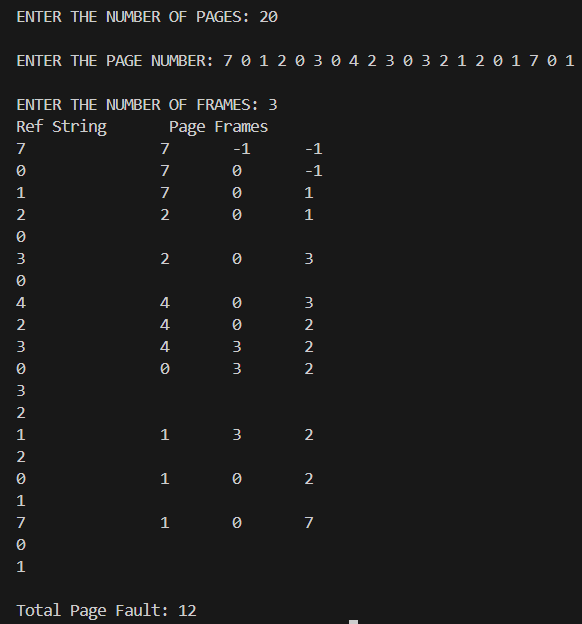
}

printf("\nTotal Page Fault: %d", count);

return 0;

}

**Output:**

****

**4) Second Chance(Clock)**

**Code:**

#include <stdio.h>

#define MAX\_FRAMES 10

int main() {

int i, j, n, pages[50], frames[MAX\_FRAMES], referenceBit[MAX\_FRAMES];

int numFrames, pageFaults = 0, pointer = 0;

printf("\nENTER THE NUMBER OF PAGES: ");

scanf("%d", &n);

printf("\nENTER THE PAGE NUMBER: ");

for (i = 0; i < n; i++)

scanf("%d", &pages[i]);

printf("\nENTER THE NUMBER OF FRAMES: ");

scanf("%d", &numFrames);

for (i = 0; i < numFrames; i++) {

frames[i] = -1;

referenceBit[i] = 0;

}

printf("Ref String\t Page Frames\t Reference Bits\n");

for (i = 0; i < n; i++) {

printf("%d\t\t", pages[i]);

int found = 0;

// Check if page is already in frames

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

if (frames[j] == pages[i]) {

found = 1; // Page hit

referenceBit[j] = 1; // Set reference bit

break;

}

}

if (!found) { // Page fault

while (1) {

if (referenceBit[pointer] == 0) {

// Replace page

frames[pointer] = pages[i];

referenceBit[pointer] = 0; // Reset reference bit

pointer = (pointer + 1) % numFrames; // Move pointer

pageFaults++;

break;

} else {

// Give second chance

referenceBit[pointer] = 0; // Reset reference bit

pointer = (pointer + 1) % numFrames; // Move pointer

}

}

}

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

if (frames[j] != -1) {

printf("%d\t", frames[j]);

} else {

printf("-\t");

}

}

printf("\t\t");

// Print the reference bits

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

printf("%d\t", referenceBit[j]);

}

printf("\n");

}

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

return 0;

}

**Output:**

