**EXPERIMENT – 33**

33. Construct a C program to simulate the optimal paging technique of memory management

#include <stdio.h>

#define MAX\_FRAMES 10

#define MAX\_REFERENCES 100

int isInFrame(int frames[], int frameCount, int page) {

for (int i = 0; i < frameCount; i++) {

if (frames[i] == page)

return 1;

}

return 0;

}

int predict(int pages[], int frames[], int n, int index, int frameCount) {

int farthest = index, pos = -1;

for (int i = 0; i < frameCount; i++) {

int j;

for (j = index; j < n; j++) {

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

if (j > farthest) {

farthest = j;

pos = i;

}

break;

}

}

if (j == n)

return i;

}

return (pos == -1) ? 0 : pos;

}

void simulateOptimal(int pages[], int n, int frameCount) {

int frames[MAX\_FRAMES];

int pageFaults = 0, filled = 0;

for (int i = 0; i < frameCount; i++)

frames[i] = -1;

printf("Page\tFrames\t\tPage Fault\n");

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

int page = pages[i];

int fault = 0;

if (!isInFrame(frames, frameCount, page)) {

fault = 1;

pageFaults++;

if (filled < frameCount) {

frames[filled++] = page;

} else {

int pos = predict(pages, frames, n, i + 1, frameCount);

frames[pos] = page;

}

}

printf("%d\t", page);

for (int j = 0; j < frameCount; j++) {

if (frames[j] == -1)

printf("- ");

else

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

}

printf("\t%s\n", fault ? "Yes" : "No");

}

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

}

int main() {

int pages[MAX\_REFERENCES], n, frameCount;

printf("Enter number of pages: ");

scanf("%d", &n);

printf("Enter the page reference string:\n");

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

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

printf("Enter number of frames: ");

scanf("%d", &frameCount);

simulateOptimal(pages, n, frameCount);

return 0;

}

SAMPLE INPUT:

Enter number of pages: 12

Enter the page reference string:

7 0 1 2 0 3 0 4 2 3 0 3

Enter number of frames: 3

SAMPLE OUTPUT:

Page Frames Page Fault

7 7 - - Yes

0 7 0 - Yes

1 7 0 1 Yes

2 2 0 1 Yes

0 2 0 1 No

3 2 0 3 Yes

0 2 0 3 No

4 4 0 3 Yes

2 4 2 3 Yes

3 4 2 3 No

0 0 2 3 Yes

3 0 2 3 No

Total Page Faults: 9