#include <stdio.h>

int findOptimal(int pages[], int n, int frames[], int m, int index) {

int res = -1, farthest = index;

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

int j;

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

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

if (j > farthest) {

farthest = j;

res = i;

}

break;

}

}

if (j == n)

return i;

}

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

}

int pageFaults(int pages[], int n, int framesCount) {

int frames[framesCount], faults = 0;

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

frames[i] = -1;

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

int j;

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

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

break;

}

}

if (j == framesCount) {

int pos = findOptimal(pages, n, frames, framesCount, i + 1);

frames[pos] = pages[i];

faults++;

}

}

return faults;

}

int main() {

int pageFrames = 3;

int pageSequence[] = {4, 1, 2, 4, 3, 2, 1, 5};

int sequenceLength = sizeof(pageSequence) / sizeof(pageSequence[0]);

int pageFaultCount = pageFaults(pageSequence, sequenceLength, pageFrames);

printf("Number of page faults: %d\n", pageFaultCount);

return 0;

}

OUTPUT

Number of page faults: 5