14. Write a C program to illustrate the page replacement method where the page which is not in demand for the longest future time is replaced by the new page and determine the number of page faults for the following test case:

No. of page frames: 3; Page reference sequence 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0 and 1.

#include <stdio.h>

int main() {

int frames[3] = {-1, -1, -1}; // Initializing frames to -1

int page\_faults = 0; // Initializing page faults to 0

int ref\_string[] = {7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1};

int n = sizeof(ref\_string)/sizeof(ref\_string[0]); // Size of reference string

int frame\_count = 0; // Counter to keep track of available frames

// Loop through the reference string

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

int page = ref\_string[i];

int flag = 0; // Flag to check if page is already in frames

// Check if page is already in frames

for(int j=0; j<3; j++) {

if(frames[j] == page) {

flag = 1;

break;

}

}

// If page is not in frames, replace the page which is not in demand for the longest future time

if(flag == 0) {

if(frame\_count < 3) {

frames[frame\_count] = page;

frame\_count++;

}

else {

int max\_future = -1;

int replace\_page\_index;

for(int j=0; j<3; j++) {

int k;

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

if(frames[j] == ref\_string[k]) {

if(k > max\_future) {

max\_future = k;

replace\_page\_index = j;

}

break;

}

}

if(k == n) {

replace\_page\_index = j;

break;

}

}

frames[replace\_page\_index] = page;

}

page\_faults++;

}

// Print current state of frames

printf("Frames: ");

for(int j=0; j<3; j++) {

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

printf("X ");

}

else {

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

}

}

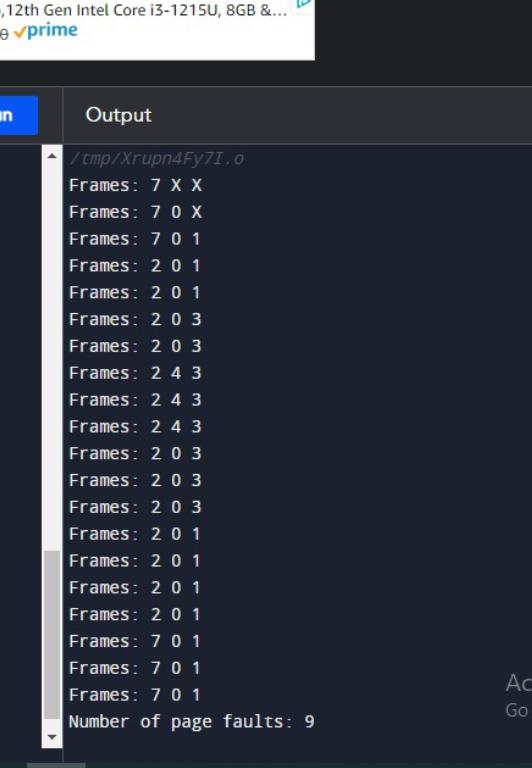
printf("\n");

}

// Print number of page faults

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

return 0;

}