7. Write a C program to illustrate the page replacement method where the current least recently used element is replaced and determine the number of page faults for the following test case:

No. of page frames: 3; Page reference sequence 1,2,3,2,1,5,2,1,6,2,5,6,3,1,3,6,1,2,4 and 3.

Program:

#include <stdio.h>

#define MAX\_PAGES 20

int main() {

int pageFrames, pageFaults = 0, time = 0;

int pageReferences[MAX\_PAGES], pageTable[MAX\_PAGES]; int i, j, oldestPage, oldestTime; printf("Enter the number of page frames: ");

scanf("%d", &pageFrames);

printf("Enter the page reference sequence (separated by spaces): "); for (i = 0; i < MAX\_PAGES; i++) {

if (scanf("%d", &pageReferences[i]) != 1) { break;

}

}

int numPages = i; for (i = 0; i < pageFrames; i++) { pageTable[i] = -1;

}

for (i = 0; i < numPages; i++) { int page = pageReferences[i]; int inPageTable = 0; for (j = 0; j < pageFrames; j++) { if (pageTable[j] == page) { inPageTable = 1;

break;

}

}

if (inPageTable) {

printf("Page %d is already in memory\n", page);

} else { pageFaults++;

printf("Page fault: Page %d\n", page); oldestPage = pageTable[0]; oldestTime = time; for (j = 0; j < pageFrames; j++) { if (pageTable[j] == -1) { oldestPage = pageTable[j];

break;

} else if (oldestTime > pageTable[j]) { oldestPage = pageTable[j]; oldestTime = pageTable[j];

}

}

for (j = 0; j < pageFrames; j++) { if (pageTable[j] == oldestPage) { pageTable[j] = page;

break;

}

}

}

for (j = 0; j < pageFrames; j++) { if (pageTable[j] != -1) { pageTable[j]++;

}

}

time++;

}

printf("Total page faults: %d\n", pageFaults); return 0;

}

Output:

