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.

Program:

#include <stdio.h>

#define MAX\_FRAMES 3

int main()

{

int frames[MAX\_FRAMES], pages[MAX\_FRAMES], page\_faults = 0; int page\_reference[] = {7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1}; int num\_pages = sizeof(page\_reference)/sizeof(page\_reference[0]);

int i, j, k, max\_future\_distance, page\_to\_replace;

for(i = 0; i < MAX\_FRAMES; i++)

{

frames[i] = -1; pages[i] = -1;

}

for(i = 0; i < num\_pages; i++)

{

int page\_found = 0; int page = page\_reference[i];

for(j = 0; j < MAX\_FRAMES; j++)

{

if(frames[j] == page)

{

page\_found = 1;

break;

}

}

if(page\_found == 0)

{

for(j = 0; j < MAX\_FRAMES; j++)

{

int page\_exists = 0; int future\_distance = 0;

for(k = i + 1; k < num\_pages; k++)

{

if(frames[j] == page\_reference[k])

{

page\_exists = 1; future\_distance = k - i;

break;

}

}

if(page\_exists == 0)

{

page\_faults++; frames[j] = page;

break;

}

if(future\_distance > max\_future\_distance)

{

max\_future\_distance = future\_distance;

page\_to\_replace = j;

}

}

page\_faults++;

frames[page\_to\_replace] = page;

}

}

printf("Number of Page Faults: %d", page\_faults);

}

Output:

