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

```c
#include<stdio.h>

int findLRU(int time[], int n){
int i, minimum = time[0], pos = 0;

for(i = 1; i < n; ++i){
if(time[i] < minimum){
minimum = time[i];
pos = i;
}
}
return pos;
}

int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j, pos, faults = 0;
printf("Enter number of frames: ");
scanf("%d", &no_of_frames);
printf("Enter number of pages: ");
scanf("%d", &no_of_pages);
printf("Enter reference string: ");
    for(i = 0; i < no_of_pages; ++i){
    scanf("%d", &pages[i]);
    }

for(i = 0; i < no_of_frames; ++i){
    frames[i] = -1;
    }
for(i = 0; i < no_of_pages; ++i){
    flag1 = flag2 = 0;

    for(j = 0; j < no_of_frames; ++j){
    if(frames[j] == pages[i]){
    counter++;
    time[j] = counter;
   flag1 = flag2 = 1;
   break;
   }
    }

    if(flag1 == 0){
for(j = 0; j < no_of_frames; ++j){
    if(frames[j] == -1){
    counter++;
    faults++;
    frames[j] = pages[i];
    time[j] = counter;
    flag2 = 1;
    break;
    }
    }
    }

    if(flag2 == 0){
    pos = findLRU(time, no_of_frames);
    counter++;
    faults++;
    frames[pos] = pages[i];
    time[pos] = counter;
    }

    printf("\n");

    for(j = 0; j < no_of_frames; ++j){
    printf("%d\t", frames[j]);
    }
}
printf("\n\nTotal Page Faults = %d", faults);

    return 0;
}
```

```
C:\Users\kalya\OneDrive\Desktop\7.ipc sm.exe
Enter number of frames: 3
Enter number of pages: 5
Enter reference string: 1
2
3
4
5

1       -1      -1
1       2       -1
1       2       3
4       2       3
4       5       3

Total Page Faults = 5
--------------------------------
Process exited after 16.45 seconds with return value 0
Press any key to continue . . .
```