Ex. No.: 11b)
Date:

LRU

Aim:

To write a c program to implement LRU page replacement algorithm.

Algorithm:

1: Start the process
2: Declare the size
3: Get the number of pages to be inserted
4: Get the value
5: Declare counter and stack
6: Select the least recently used page by counter value
7: Stack them according the selection.
8: Display the values
9: Stop the process

Program Code:

```
#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]; int 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 ", frames[j]);
    }
}

printf("\n\nTotal Page Faults = %d\n", faults);
return 0;


}
```

Sample Output :
Enter number of frames: 3
Enter number of pages: 6
Enter reference string: 5 7 5 6 7 3
5 -1 -1
5 7 -1
5 7 -1
5 7 6
5 7 6
3 7 6
Total Page Faults = 4

Output :

```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];
    int 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);
```

```
input
5 7 6
3 7 6

Total Page Faults = 4

...Program finished with exit code 0
Press ENTER to exit console.
```

Result:
Program executed successfully and output is verified.