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>
# include <stdlib.h>
# define MAX 100
int findLRU (int time[], int n) {
        int i, minimum = time[0], pos = 0;

        for (i=1, i<n; i++) {
            if (time[i] < maximum) {
                minimum = time[i];
                pos = i;
            }
        }
        return pos;
}
int main() {
    int frames[MAX], pages[MAX], time[MAX];
    int n, capacity, counter = 0, faults = 0;

    int i, J, pos, flag1, flag2;
```

```
Printf ("Enter the number of pages :");
Scanf ("%d", &n);

printf ("Enter the page refrence string :");
for (i=0; i<n; i++) {
    Scanf ("%d", &pages[i]));
}
printf ("Enter the no of Frames: ");
Scanf ("%d", &capacity);
for (i=0 ; i < capacity ; i++) {
        frames[i] = -1 ;
}
for (i=0; i<n ; i++) {
        flag1 = flag2 = 0;
        for ( J= 0; J < capacity ; J++) {
            if (frames[J] == pages[i]) {
            counter++ ;
            time[J] = counter++;
            flag1 = flag2 = 1;
            break;
        }
    }
    if(flag1 == 0) {
        for (J=0 ; J < capacity : J++) {
            if (frames[J] == -1) {
                counter ++ ;
                faults ++ ;
                frames[J] = pages[i];
                time[J] = counter ;
                flag2 = 1;
                break;
            }
        }
    }
}
```

```
if ( flag2 == 0) {
        pos = findLRU ( time, capacity);
        counter ++;
        faults ++;
        frames [pos] = pages [i];
        time [pos] = counter;
    }
    printf ("Memory after inserting %d :", pages [i]));
    for (J=0; J< capacity ; J++) {
        if ( frames [J] != -1)
            printf ("%d", frames [J]);
        else
            printf ("_");
    }
    printf ("Total page Faults = %d \n", faults);
    return 0;
}
```

Input :

Enter the number of pages : 12

Enter the reference string : 1 3 0 3 5 6 3 0 6 4 1 7

Enter the number of frames : 3

Output:

1 : 1 _

3 : 1 3 _

0 : 1 3 0

5 : 5 3 0

6 : 5 6 0

3 : 3 6 0

0 : 3 6 0

6 : 3 6 0

4 : 4 6 0

1 : 4 1 0

7 : 4 1 7

Total page Faults = 9

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
```

Result:

Thus, LRU page replacement algorithm is implemented and executed sucessfully.