Ex. No.: 11b)
Date: 17/4/25

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 main(){
    int refstr[100], frames[20], recent[20];
    int refsize, framesize;
    int i, j, k, time =0, Pf =0, isHit, index;
    printf("Enter the Number of pages :");
    scanf("%d", &refsize);
    for(int i=0; i<refsize; i++){
        printf("[%d]:" i+1);
        scanf("%d", &refstr[i]);
    }
    printf("Enter page frame size :");
    scanf("%d", &framesize);
    for (i=0; i< framesize; i++){
        frame[i] = -1;
        recent[i] = -1;
    }
```

```
printf ("\n");
for(int i = 0; i < refsize ; i++){
    isHit =0;
    for (int j =0; j < framesize ; j++){
        if ( frames[j] = = refstr[i]){
            isHit = 1;
            recent[j] = time++;
            break;
} }
if ( isHit){
    printf ("%2d -> NO page fault \n",
                                refstr[i]);
    continue;
}
int emplyind = -1
for (j=0; j < framesize ; j++){
        if ( frames[j] = = -1){
            emplyind =j;
            break;
        }
}
if ( emplyind ! = -1){
        frames[emplyind] = refstr[i];
        recent[emplyind] = time++;
}else{
        int min = recent[0];
        druindex =0;
```

```
for(j=1; j < framesize; j++){
        if (recent[j] < min){
                min = recent[j];
        }       truindex = j;
}
frames[truindex] = refstr[i];
recent[truindex] = time++;
}
pf++;
printf("%d" -> ", refstr[i]);
for (int k=0; k < framesize ; k++){
        if (frames[k]! = -1)
                printf("%d", frames[k]).
        }
        printf(" ⇒ page fault\n");
}
printf("\n Total page faults : %d\n", pf);

}
```

OUTPUT :

Enter number of pages : 14

Enter page frame : 4

Enter[1] = 7
Enter[2] = 0
Enter[3] = 1
Enter [4] = 2
Enter[5] = 0
Enter [6] = 3
Enter [7] = 0
Enter[8] = 4
Enter[9] = 2
Enter[10] = 3
Enter[11] = 0
Enter[12] = 3
Enter[13] = 2
Enter[14] = 3

7 → 7 ⇒ page fault
0 → 70 ⇒ page fault
1 → 701 ⇒ page fault
2 → 7012 ⇒ page fault
0 → NO page fault
4 → 3042 ⇒ page fault
2 → NO page fault
3 → NO page fault
0 → NO page fault
3 → NO page fault
2 → NO page fault
3 → NO page fault

Total page faults : 6

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:

A C program for finding the page fault using LRU page replacement technique is implemented successfully.

ACE Scanner