```
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] = -\overline{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 < \text{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
57-1
57-1
576
576
3 7 6
Total Page Faults = 4
```

Output:

```
int findLRU(int ine [], int n) {
  int i, minimum = vim [0], pos = 0;
  for(i = 1; i < n; ++i) {
    if(i = [i] < minimum) {
        minimum = vim [i];
        pos = i;
    }
}</pre>

}
10
}
11 return pos;
12
}
13
14 int main() {
15 int no_of_frames, no_of_pages, frames[10], pages[30], counter = 0, lime[10];
16 int flag1, flag2, i, j, pos, faults = 0;
17
18 print("Enter number of frames: ");
19 stont("%d", &no_of_frames);
20
21 print("Enter number of pages: ");
22 stont("%d", &no_of_pages);
23 print("%d", &no_of_pages);
24 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
26 print("%d", &no_of_pages);
27 print("%d", &no_of_pages);
28 print("%d", &no_of_pages);
29 print("%d", &no_of_pages);
20 print("%d", &no_of_pages);
20 print("%d", &no_of_pages);
21 print("%d", &no_of_pages);
22 print("%d", &no_of_pages);
23 print("%d", &no_of_pages);
24 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
26 print("%d", &no_of_pages);
27 print("%d", &no_of_pages);
28 print("%d", &no_of_pages);
29 print("%d", &no_of_pages);
20 print("%d", &no_of_pages);
21 print("%d", &no_of_pages);
22 print("%d", &no_of_pages);
23 print("%d", &no_of_pages);
24 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
26 print("%d", &no_of_pages);
27 print("%d", &no_of_pages);
28 print("%d", &no_of_pages);
29 print("%d", &no_of_pages);
20 print("%d", &no_of_pages);
21 print("%d", &no_of_pages);
22 print("%d", &no_of_pages);
23 print("%d", &no_of_pages);
24 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
26 print("%d", &no_of_pages);
27 print("%d", &no_of_pages);
28 print("%d", &no_of_pages);
29 print("%d", &no_of_pages);
20 print("%d", &no_of_pages);
21 print("%d", &no_of_pages);
22 print("%d", &no_of_pages);
23 print("%d", &no_of_pages);
24 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
26 print("%d", &no_of_pages);
27 print("%d", &no_of_pages);
28 print("%d", &no_of_pages);
29 print("%d", &no_of_pages);
20 print("%d", &no_of_pages);
21 print("%d", &no_of_pages);
21 print("%d", &no_of_pages);
22 print("%d", &no_of_pages);
22 print("%d", &no_of_pages);
23 print("%d", &no_of_pages);
24 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
25 print("%d", &no_of_pages);
25 print("%d", &no
                                                                                                                                                 ("Enter number of pages: ");
"("%d", &no_of_pages);
y / I□ ☆ Ŋ
5 7 6
5 7 6
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          input
       otal Page Faults = 4
                      Program finished with exit code 0
```

Result:

Program executed successfully and output is verified.