**Practical 6**

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

**Theory:** LRU stands for Least Recently Used. As the name suggests, this algorithm is based on the strategy that whenever a page fault occurs, the least recently used page will be replaced with a new page. So, the page not utilized for the longest time in the memory (compared to all other pages) gets replaced. This strategy is known as LRU paging.

A page fault occurs when a running program tries to access a piece (or page) of memory that is not already present in the main memory (RAM). On the other hand, if that page is already present in the memory, it is called a page hit. The LRU page replacement algorithm comes into the picture whenever a page fault occurs.

**Advantages of LRU Page Replacement Algorithm:**

The page that has not been used for the longest time gets replaced. It gives lesser page faults than any other algorithm. The algorithm is capable of complete analysis.

**Disadvantages of LRU Page Replacement Algorithm:**

The execution of this algorithm is complicated. This algorithm's execution may require significant assistance from the hardware.

**Example :** In Least Recently Used (LRU) algorithm is a Greedy algorithm where the page to be replaced is least recently used. The idea is based on locality of reference, the least recently used page is not likely Let say the page reference string 7 0 1 2 0 3 0 4 2 3 0 3 2 . Initially we have 4 page slots empty.

Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> 4 Page faults

0 is already their so —> 0 Page fault.

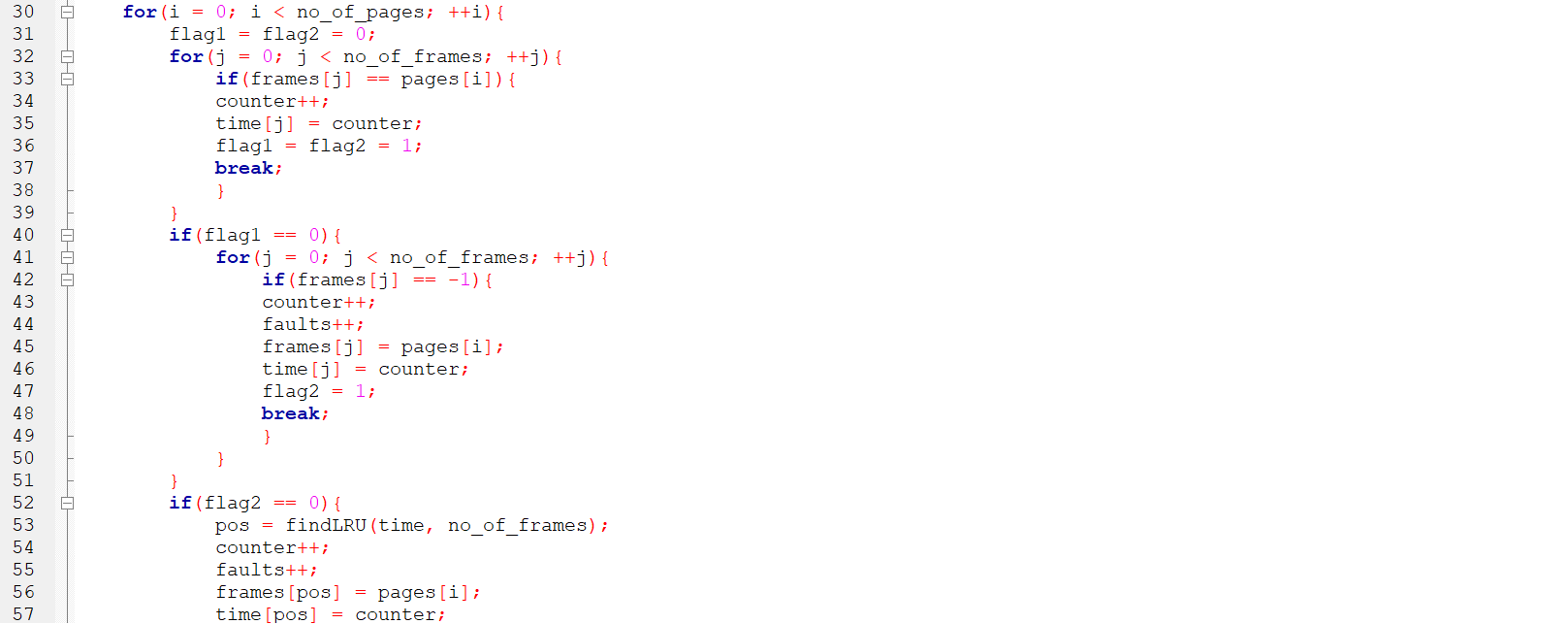
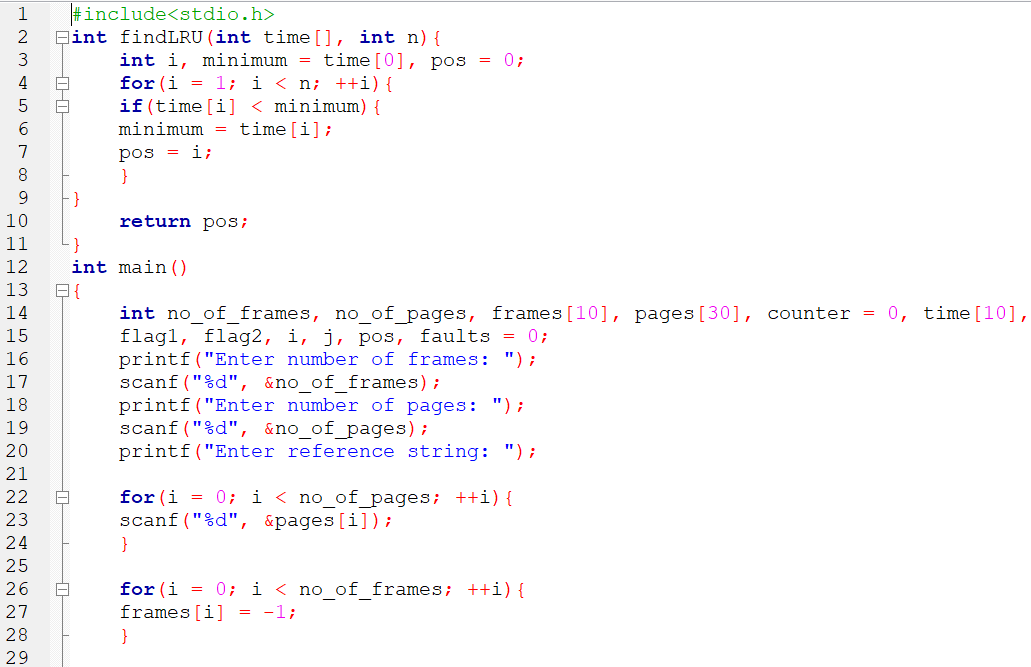
when 3 came it will take the place of 7 because it is least recently used —>1 Page fault

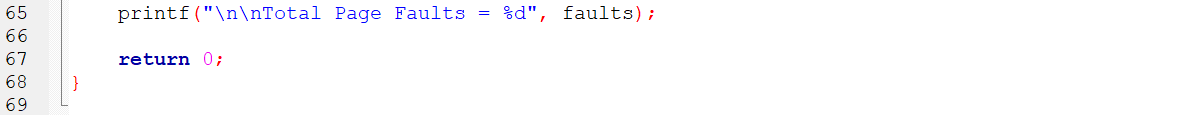
0 is already in memory so —> 0 Page fault.

4 will takes place of 1 —> 1 Page Fault

Now for the further page reference string —> 0 Page fault because they are already available in the memory.

**Program:**

****

****