**32.Construct a C program to simulate the Least Recently Used paging technique of memory management.**

**Aim:** To Construct a C program to simulate the Least Recently Used paging technique of memory management.

# **Algorithm:**

* 1. Create an array to represent the page frames in memory.
  2. Initialize all page frames to -1, indicating that they are empty.
  3. Create a queue or a data structure (e.g., a doubly-linked list) to maintain the order of pages based on their usage history.
  4. Initialize a counter for page hits and page faults to zero.
  5. Read the reference string (sequence of page numbers) from the user or use a predefined array.
  6. or each page in the reference string, do the following:
  7. Check if the page is already in memory (a page hit).
  8. If it's a page hit, update the position of the page in the usage history data structure to indicate it was recently used.
  9. If it's a page fault (page not in memory), do the following:
  10. Increment the page fault count.
  11. Find the least recently used page in the usage history data structure (e.g., the front of the queue or the tail of the list).
  12. Remove the least recently used page from memory and the usage history.
  13. Load the new page into memory and add it to the back of the usage history.
  14. Update the display to show the page replacement. 15.Continue this process for all pages in the reference string.

16.After processing all pages, display the total number of page faults

**Program:**

#include<stdio.h>

#include<conio.h>

main()

{

int i, j , k, min, rs[25], m[10], count[10], flag[25], n, f, pf=0, next=1; printf("Enter the length of reference string -- ");

scanf("%d",&n);

printf("Enter the reference string -- "); for(i=0;i<n;i++)

{

scanf("%d",&rs[i]); flag[i]=0;

}

printf("Enter the number of frames -- "); scanf("%d",&f);

for(i=0;i<f;i++){

count[i]=0; m[i]=-1;

}

printf("\nThe Page Replacement process is -- \n"); for(i=0;i<n;i++)

{

for(j=0;j<f;j++)

{

if(m[j]==rs[i])

{

flag[i]=1; count[j]=next; next++;

}

}

if(flag[i]==0)

{

if(i<f)

{ m[i]=rs[i]; count[i]=next; next++;

}

else

{ min=0; for(j=1;j<f;j++) if(count[min] > count[j]) min=j;

m[min]=rs[i]; count[min]=next; next++;

}

pf++;

}

for(j=0;j<f;j++) printf("%d\t", m[j]); if(flag[i]==0)

printf("PF No. -- %d" , pf); printf("\n");

}

printf("\nThe number of page faults using LRU are %d",pf); getch();

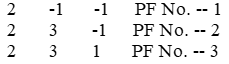
}

# **Output:**

Enter the length of reference string -- 3 Enter the reference string -- 2

3

1

Enter the number of frames -- 3 The Page Replacement process is --

The number of page faults using LRU are 3