# **Construct a C program to simulate the First in First Out paging technique of memory management.**

**Aim:** To Construct a C program to simulate the First in First Out 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. Initialize a queue to keep track of the order in which pages are loaded into memory.
  4. Initialize variables 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. For 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 display and move to the next page. 10.If it's a page fault (page not in memory)
  9. Increment the page fault count.
  10. Remove the oldest page in memory (the one at the front of the queue).
  11. Load the new page into the memory and enqueue it.
  12. Update the display to show the page replacement
  13. Continue this process for all pages in the reference string.
  14. After processing all pages, display the total number of page faults

# **Program:**

#include<stdio.h> #include<conio.h> main()

{

int i, j, k, f, pf=0, count=0, rs[25], m[10], n; printf("\n Enter the length of reference string -- "); scanf("%d",&n);

printf("\n Enter the reference string -- ");

for(i=0;i<n;i++)

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

printf("\n Enter no. of frames -- "); scanf("%d",&f);

for(i=0;i<f;i++) m[i]=-1;

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

{

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

{

if(m[k]==rs[i]) break;

}

if(k==f)

{

m[count++]=rs[i]; pf++;

}

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

printf("\tPF No. %d",pf);

printf("\n"); if(count==f) count=0;

}

printf("\n The number of Page Faults using FIFO are %d",pf); getch();

}

# **Output:**

