**Program VIII: Design, develop and implement page replacement using FIFO algorithm. Assume suitable input required to demonstrate the results.**

**Page Replacement Algorithms:**

* Page replacement is basic to demand paging. It completes the separation between logical memory and physical memory. With this mechanism, an enormous virtual memory can be provided for programmers on a smaller physical memory.
* There are many different page-replacement algorithms. Every operating system probably has its own replacement scheme.
* A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen.
* If the recent past is used as an approximation of the near future, then the page that has not been used for the longest period of time can be replaced. This approach is the Least Recently Used (LRU) algorithm.
* LRU replacement associates with each page the time of that page's last use. When a page must be replaced, LRU chooses the page that has not been used for the longest period of time.
* Least frequently used (LFU) page-replacement algorithm requires that the page with the smallest count be replaced. The reason for this selection is that an actively used page should have a large reference count.

**ALGORITHM:**

Step 1: Start the program

Step 2: Read the number of frames

Step 3: Read the number of pages

Step 4: Read the page numbers

Step 5: Initialize the values in frames to -1

Step 6: Allocate the pages in to frames in First in first out order.

Step 7: Display the number of page faults.

Step 8: Stop the program

**PROGRAM:**

**SOURCE CODE:**

/\* A program to simulate FIFO Page Replacement Algorithm \*/

#include<stdio.h>

int main()

{

int a[5],b[20],n,p=0,q=0,m=0,h,k,i,q1=1;

char f='F';

printf("Enter the Number of Pages:");

scanf("%d",&n);

printf("Enter %d Page Numbers:",n);

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

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

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

{

if(p==0)

{

if(q>=3)

q=0;

a[q]=b[i];

q++;

if(q1<3)

{

q1=q;

}

}

printf("\n%d",b[i]);

printf("\t");

for(h=0;h<q1;h++)

printf("%d",a[h]);

if((p==0)&&(q<=3))

{

printf("-->%c",f);

m++;

}

p=0;

for(k=0;k<q1;k++)

{

if(b[i+1]==a[k])

p=1;

}

}

printf("\nNo of faults:%d",m);

}

**Program IX: Design, develop and implement page replacement using LRU algorithms. Assume suitable input required to demonstrate the results.**

**ALGORITHM**:

Step 1: Start the program.

Step 2: Read the number of frames.

Step 3: Read the number of pages.

Step 4: Read the page numbers.

Step 5: Initialize the values in frames to -1.

Step 6: Allocate the pages in to frames by selecting the page that has not been used for the longest

period of time.

Step 7: Display the number of page faults.

Step 8: Stop the program.

**PROGRAM :**

**SOURCE CODE:**

/\* A program to simulate LRU Page Replacement Algorithm \*/

#include<stdio.h>

int main()

{

int a[5],b[20],p=0,q=0,m=0,h,k,i,q1=1,j,u,n;

char f='F';

printf("Enter the number of pages:");

scanf("%d",&n);

printf("Enter %d Page Numbers:",n);

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

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

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

{

if(p==0)

{

if(q>=3)

q=0;

a[q]=b[i];

q++;

if(q1<3)

{

q1=q;

}

}

printf("\n%d",b[i]);

printf("\t");

for(h=0;h<q1;h++)

printf("%d",a[h]);

if((p==0)&&(q<=3))

{

printf("-->%c",f);

m++;

}

p=0;

if(q1==3)

{

for(k=0;k<q1;k++)

{

if(b[i+1]==a[k])

p=1;

}

for(j=0;j<q1;j++)

{

u=0;

k=i;

while(k>=(i-1)&&(k>=0))

{

if(b[k]==a[j])

u++;

k--;

}

if(u==0)

q=j;

}

}

else

{

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

{

if(b[i+1]==a[k])

p=1;

}

}

}

printf("\nNo of faults:%d",m);

}