**Lab Report No: 11**

**Name of the lab report :** Implementation of FIFO page replacement Algorithm.

**Name: Binodon**

**ID: IT-17046**

**Objective:** FIFO page replacementalgorithm Definition & executable code in codeblocks.

**Q.1 What is FIFO page replacement algorithm?**

**Ans:**This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal

**Q.2 How to implemented in C?**

**Ans:**

#include<stdio.h>

int main()

{

inti,j,n,a[50],frame[10],no,k,avail,count=0;

printf("\n ENTER THE NUMBER OF PAGES:\n");

scanf("%d",&n);

printf("\n ENTER THE PAGE NUMBER :\n");

for(i=1;i<=n;i++)

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

printf("\n ENTER THE NUMBER OF FRAMES :");

scanf("%d",&no);

for(i=0;i<no;i++)

frame[i]= -1;

j=0;

printf("\tref string\t page frames\n");

for(i=1;i<=n;i++)

{

printf("%d\t\t",a[i]);

avail=0;

for(k=0;k<no;k++)

if(frame[k]==a[i])

avail=1;

if (avail==0)

{

frame[j]=a[i];

j=(j+1)%no;

count++;

for(k=0;k<no;k++)

printf("%d\t",frame[k]);

}

printf("\n");

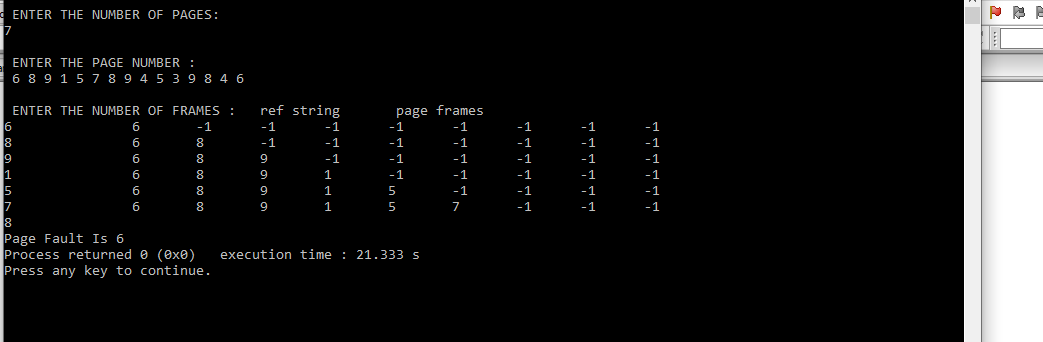
}

printf("Page Fault Is %d",count);

return 0;

}

**Output:**

****