**Experiment No. 14**

**Experiment Name:** Implementation of FIFO page replacement algorithm .

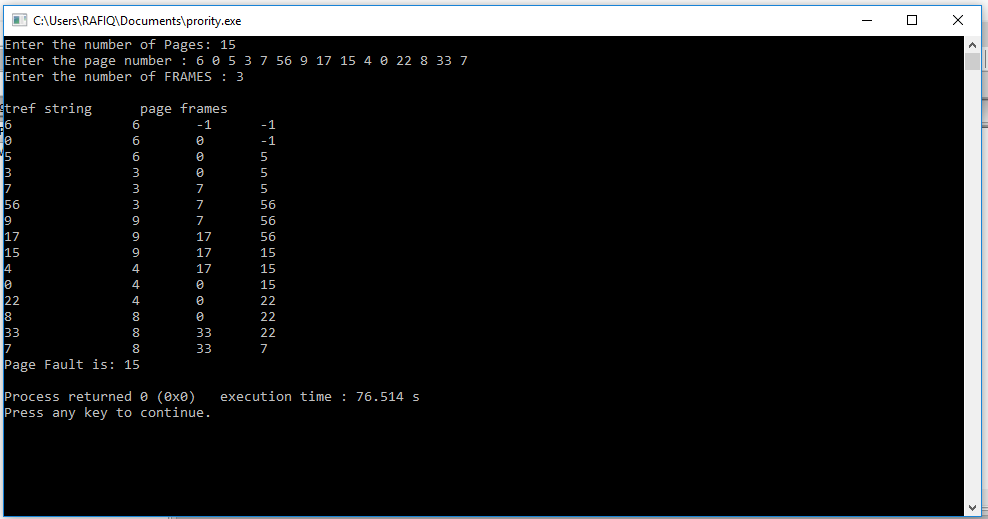
**Aim and Objectives:**

* What is FIFO page replacement algorithm.
* How to implementation this algorithm

**Source Code:**

|  |
| --- |
| #include<stdio.h>  int main()  {  int i,j,n,a[50],frame[10],no,k,avail,count=0;  printf("Enter the number of Pages: ");  scanf("%d",&n);  printf("Enter the page number : ");  for(i=1; i<=n; i++)  scanf("%d",&a[i]);  printf("Enter the number of FRAMES : ");  scanf("%d",&no);  for(i=0; i<no; i++)  frame[i]= -1;  j=0;  printf("\n");  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);  printf("\n");  return 0;  } |

**Output:**



**Conclusion:** In FIFO page replacement algorithm- for each page we track the time when it was brought into the memory and when any replacement request comes then oldest page is chosen. If we choose a queue to hold all pages in memory then its more easy to understand and implement rather than tracking time of all pages.