Experiment 6

**AIM** : To implement First In First Out (FIFO) page replacement algorithm **.**

**THEORY** :

In operating systems that use paging for memory management, page

replacement algorithms are needed to decide which page needs to be

replaced when a new page comes in. Whenever a new page is referred to

and not present in memory, page fault occurs and the Operating System

replaces one of the existing pages with a newly needed page. Different

page replacement algorithms suggest different ways to decide which page

to replace. The target for all algorithms is to reduce the number of page

faults.

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, the page in the front of the queue is selected for

removal.

**CODE:**

**FIFO :**

#include <bits/stdc++.h>

using namespace std;

const int N=100;

int n;

int frame\_size;

int pages[N];

int flag[N];

void printQueue(queue<int> q)

{

while (!q.empty()){

cout<<" "<<q.front();

q.pop();

}

cout<<endl;

}

void fifo(void)

{

queue<int> Q;

int page\_faults=0;

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

{

if(flag[pages[i]]==true)

{

printQueue(Q);

}

else

{

Q.push(pages[i]);

flag[pages[i]]=true;

if(Q.size()>frame\_size)

{

int p= Q.front();

flag[p]=false;

Q.pop();

}

page\_faults++;

cout<<"Page Fault";

printQueue(Q);

}

}

cout<<"\nTotal Page Faults: "<<page\_faults;

return;

}

int main()

{

cout<<"Number of Frames: ";

cin>>frame\_size;

cout<<"Page Reference Stream Length: ";

cin>>n;

cout<<"Page Reference Stream:\n";

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

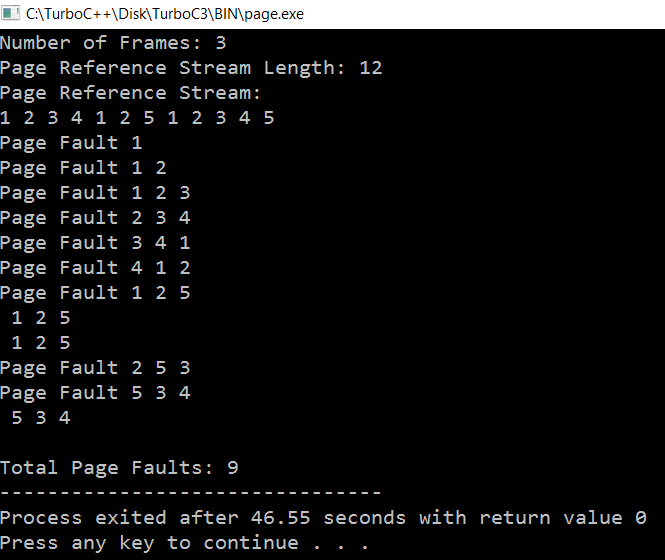
cin>>pages[i];

fifo();

return 0;

}

**OUTPUT :**

****

**CONCLUSION:** Hence we have successfully implemented First-in-First

-Out scheduling algorithm.