**Experiment No:** 05

**Experiment Name:**Implementation of Optimal Page Replacement algorithm.

**Objectives:** Learn about Optimal page replacement algorithm. Emplement Optimal page replacement algorithm by using c program. And testing the program different input and find output .

**CODE:**

#include<stdio.h>

void main()

{

int nop,nof,page[20],i,count=0;

printf("\nEnter the No. of Pages: ");

scanf("%d",&nop);

//Store the no of pages

printf("\n Enter the Reference String:\n");

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

{

scanf("%d",&page[i]); //Array for Storing Reference String

}

printf("\n Enter the No of frames: ");

scanf("%d",&nof);

int frame[nof],fcount[nof];

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

{

frame[i]=-1; //Frame Array

fcount[i]=0; // Track the next Availability of frames

}

i=0;

while(i<nop)

{

int j=0,flag=0;

while(j<nof)

{

if(page[i]==frame[j]) // Checking Whether the Page is Already in frame or not

{

flag=1;

}

j++;

}

j=0;

printf("\n");

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

if(flag==0)

{

if(i>=nof)

{

int max=0,k=0;

while(k<nof)

{

int dist=0,j1=i+1;

while(j1<nop)

{

if(frame[k]!=page[j1]) //Calculating Distances of pages that are in the frame to their next occurence

dist++;

else

{

break;

}

j1++;

}

fcount[k]=dist; //Storing Distances into array

k++;

}

k=0;

while(k<nof-1)

{

if(fcount[max]<fcount[k+1]) //Finding out the maxximum distance

max=k+1;

k++;

}

frame[max]=page[i];

}

else

{

frame[i%nof]=page[i];

}

count++; // Increasing Page Fault.

while(j<nof)

{

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

j++;

}

}

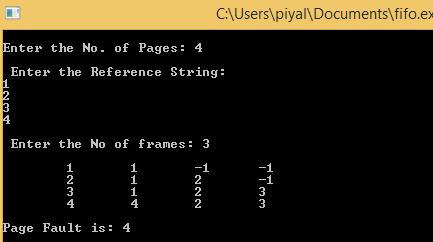
i++;

}

printf("\n\nPage Fault is: %d\n",count);

}

**Output:**



**Conclusion:** In this lab experiment, we learn about Optimal page replacement algorithm. We also learn how to implement Optimal page replacement by using C program And testing the program different input and find output.We get proper output.