

```

#include<stdio.h>
int np,nf,pf,page[50];
int frams[50][10];

int fifo()
{
    int i,j,f,k=0;
    for(i=0;i<nf;i++)
        frams[0][i]=-99;
    for(i=1;i<=np;i++)
    {
        f=0;
        for(j=0;j<nf;j++)
        {
            frams[i][j]=frams[i-1][j];
            if(page[i]==frams[i-1][j])
            {
                f=1;
            }
        }
    }

    if(f==0)
    {
        frams[i][k]=page[i];
        k++;
        pf++;
    }
    if(k==nf)
        k=0;
}

output();
pf=0;
}

// LRU
int lru()
{
    int f,flag[100],eflag,count[100];
    int k=0,i,j,max,index;
    //TO EMPTY FRAM
    for(i=0;i<nf;i++)
    {
        frams[0][i]=-99;
    }
    for(i=1;i<=np;i++)
    {
        f=0,eflag=0;
        for(j=0;j<nf;j++)
        {
            frams[i][j]=frams[i-1][j]; //TO COPY FIRST BLOCK IN NEXT

            if(page[i]==frams[i-1][j])
            {
                f=1; //For page exit
            }

            if(frams[i-1][j]==-99)
            {

```

```

eflag=1; //For Empty Frame
}
count[j]=0;
flag[j]=0;
}
//copy in empty fram

if((f==0)&&(eflag==1))
{
frames[i][k]=page[i];
k++;
pf++;
}
else if(f==0)
{
//LRU PAGE
for(k=i-1;k>0;k--)
{
for(j=0;j<nf;j++)
{
if(frames[i-1][j]==page[k])
flag[j]=1;
else if(flag[j]!=1)
count[j]++;
}
}

max=0;

//find max count
for(j=0;j<nf;j++)
{
if(max<count[j])
{
max=count[j];
index=j;
}

}

frames[i][index]=page[i];
pf++;

}
}
output();
pf=0;
}

int optimal()
{
int f,flag[100],eflag,count[100];
int k=0,i,j,max,index;
//TO EMPTY FRAM
for(i=0;i<nf;i++)
{
frames[0][i]=-99;
}
for(i=1;i<=np;i++)

```

```

{
f=0,eflag=0;
for(j=0;j<nf;j++)
{
frames[i][j]=frames[i-1][j]; //TO COPY FIRST BLOCK IN NEXT

if(page[i]==frames[i-1][j])
{
f=1; //For page exit
}

if(frames[i-1][j]==-99)
{
eflag=1; //For Empty Frame
}
count[j]=0;
flag[j]=0;
}
//copy in empty fram

if((f==0)&&(eflag==1))
{
frames[i][k]=page[i];
k++;
pf++;
}
else if(f==0)
{
//OPTIMAL PAGE
for(k=i+1;k<=np;k++)
{
for(j=0;j<nf;j++)
{
if(frames[i-1][j]==page[k])
flag[j]=1;
else if(flag[j]!=1)
count[j]++;
}
}
max=0;

//find max count
for(j=0;j<nf;j++)
{
if(max<count[j])
{
max=count[j];
index=j;
}
}

frames[i][index]=page[i];
pf++;

}
}
output();
pf=0;

```

```

}

int output()
{
    int i,j;
    for(i=1;i<=np;i++)
    {
        printf("\n\n");
        for(j=0;j<nf;j++)
        {
            if(frams[i][j]==-99)
                printf(" ");
            else
                printf(" %d",frams[i][j]);
        }
    }
    printf("\n PAGE FAULT:");

    printf("%d",pf);
}

int main()
{
    int i,ch;

    printf("\n Enter NU_OF Pages:");
    scanf("%d",&np);

    printf("\n Enter Page Sequince FOR %d pages:",np);
    for(i=1;i<=np;i++)
    {
        scanf("%d",&page[i]);
    }
    printf("\n ENter NU_offrame:");
    scanf("%d",&nf);

    while(1)
    {
        printf("\n 1:FIFO \n 2:LRU \n 3:OPTIMAL \n 4:EXIT");
        printf("\n Enter your choic:");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1:fifo();
            break;
            case 2:lru();
            break;
            case 3: optimal();
            break;
            case 4:exit(0);
            break;
        }
    }
}

```

```
OUTPUT:-  
[root@localhost Documents]# gcc pagerepl.c  
[root@localhost Documents]# ./a.out
```

Enter NU_OF Pages:10

Enter Page Sequence FOR 10 pages:7 0 1 2 0 3 0 4 3 1

ENter NU_offrame:3

1:FIFO

2:LRU

3:OPTIMAL

4:EXIT

Enter your choic:1

7

7 0

7 0 1

2 0 1

2 0 1

2 3 1

2 3 0

4 3 0

4 3 0

4 1 0

PAGE FAULT:8
1:FIFO
2:LRU
3:OPTIMAL
4:EXIT
Enter your choic:2

7

7 0

7 0 1

2 0 1

2 0 1

2 0 3

2 0 3

4 0 3

4 0 3

4 1 3

PAGE FAULT:7

1:FIFO

2:LRU

3:OPTIMAL

4:EXIT

Enter your choic:3

7

7 0

7 0 1

2 0 1

2 0 1

3 0 1

3 0 1

3 4 1

3 4 1

3 4 1

PAGE FAULT:6

1:FIFO

2:LRU

3:OPTIMAL

4:EXIT

```
Enter your choic:4  
[root@localhost Documents]#
```