

## Program :

```
#include<stdio.h>
int n,nf;
int in[100];
int p[50];
int hit=0;
int i,j,k;
int pgfaultcnt=0;
void getData()
{
    printf("\nEnter length of page reference sequence:"); scanf("%d",&n);
    printf("\nEnter the page reference sequence:"); for(i=0; i<n; i++)
        scanf("%d",&in[i]);
    printf("\nEnter no of frames:");
    scanf("%d",&nf);
}
void initialize()
{
    pgfaultcnt=0;
    for(i=0; i<nf; i++)
        p[i]=9999;
}
int isHit(int data)
{
    hit=0;
    for(j=0; j<nf; j++)
    {
        if(p[j]==data)
        {
            hit=1;
            break;
        }
    }
    return hit;
}
int getHitIndex(int data)
{
    int hitind;
    for(k=0; k<nf; k++)
    {
        if(p[k]==data)
        {
            hitind=k;
            break;
        }
    }

    return hitind;
}
void dispPages()
{
    for (k=0; k<nf; k++)
    {
        if(p[k]!=9999)
            printf(" %d",p[k]);
    }
}
void dispPgFaultCnt()
{
    printf("\nTotal no of page faults:%d",pgfaultcnt); }
```

```

void fifo()
{
    initialize();
    for(i=0; i<n; i++)
    {
        printf("\nFor %d :",in[i]);
        if(isHit(in[i])==0)
        {
            for(k=0; k<nf-1; k++)
            p[k]=p[k+1];
            p[k]=in[i];
            pgfaultcnt++;
            dispPages();
        }
        else
        printf("No page fault");
    }
    dispPgFaultCnt();
}

void optimal()
{
    initialize();
    int near[50];
    for(i=0; i<n; i++)
    {
        printf("\nFor %d :",in[i]);
        if(isHit(in[i])==0)
        {
            for(j=0; j<nf; j++)
            {
                int pg=p[j];
                int found=0;
                for(k=i; k<n; k++)
                {
                    if(pg==in[k]) {
                        near[j]=k; found=1;
                        break; }
                }
            }
            else
            found=0; }
        if(!found)
            near[j]=9999; }
        int max=-9999;
        int repindex;
        for(j=0; j<nf; j++) {
            if(near[j]>max) {
                max=near[j]; repindex=j;
            }
        }
        p[repindex]=in[i]; pgfaultcnt++;
        dispPages();
    }
    else
        printf("No page fault"); }
    dispPgFaultCnt();
}

void lru()
{
    initialize();
    int least[50];
    for(i=0; i<n; i++)
    {
        printf("\nFor %d :",in[i]);

```

```

if(isHit(in[i])==0)
{
for(j=0; j<nf; j++) {
int pg=p[j]; int found=0; for(k=i-1; k>=0; k--) {
if(pg==in[k]) {
least[j]=k; found=1;
break;
}
}
else
found=0; }
if(!found)
least[j]=-9999; }
int min=9999;
int repindex;
for(j=0; j<nf; j++)
{
if(least[j]<min) {
min=least[j]; repindex=j;
}
}
p[repindex]=in[i];
pgfaultcnt++;
dispPages();
}
else 9
printf("No page fault!"); }
dispPgFaultCnt();
}
int main()
{
int choice;
while(1)
{
printf("\nPage Replacement Algorithms\n1.Enter data\n2.FIFO\n3.Optimal\n4.LRU\n5.Exit\n\nEnter your choice:");
scanf("%d",&choice);
switch(choice)

{
case 1: getData();
break;
case 2: fifo();
break;
case 3:optimal();
break;
case 4: lru();
break;
default: return 0;
break;
}
}
}
}

```

# Output :

```
admin1@admin1-M5-7D48: ~/Desktop/Niraj/Ass6
admin1@admin1-M5-7D48:~/Desktop/Niraj/Ass6$ touch Ass6a.c
admin1@admin1-M5-7D48:~/Desktop/Niraj/Ass6$ gcc Ass6.c
admin1@admin1-M5-7D48:~/Desktop/Niraj/Ass6$ ./a.out

Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:1

Enter length of page reference sequence:15

Enter the page reference sequence:5 0 2 3 0 1 3 4 5 4 2 0 3 4 3

Enter no of frames:3

Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:2

For 5 : 5
For 0 : 5 0
For 2 : 5 0 2
For 3 : 0 2 3
For 0 :No page fault
For 1 : 2 3 1
For 3 :No page fault
For 4 : 3 1 4
For 5 : 1 4 5
For 4 :No page fault
For 2 : 4 5 2
For 0 : 5 2 0
For 3 : 2 0 3
For 4 : 0 3 4
For 3 :No page fault
Total no of page faults:11
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
```

```
5.Exit
Enter your choice:3

For 5 : 5
For 0 : 5 0
For 2 : 5 0 2
For 3 : 5 0 3
For 0 :No page fault
For 1 : 5 1 3
For 3 :No page fault
For 4 : 5 4 3
For 5 :No page fault
For 4 :No page fault
For 2 : 2 4 3
For 0 : 0 4 3
For 3 :No page fault
For 4 :No page fault
For 3 :No page fault
Total no of page faults:8
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:4

For 5 : 5
For 0 : 5 0
For 2 : 5 0 2
For 3 : 3 0 2
For 0 :No page fault!
For 1 : 3 0 1
For 3 :No page fault!
For 4 : 3 4 1
For 5 : 3 4 5
For 4 :No page fault!
For 2 : 2 4 5
For 0 : 2 4 0
For 3 : 2 3 0
For 4 : 4 3 0
For 3 :No page fault!
Total no of page faults:11
Page Replacement Algorithms
1.Enter data
2.FIFO
```