Program-7
Write a C program to simulate page replacement algorithms. a) FIFO b) LRU c) Optimal
Code:

```
#include<stdio.h>
int
n,
f,
i,
j,
k;
int
in[
10
0];
int
p[
50
];
int
hit
=0
;
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",&f);
}

void initialize()
{

pgfaultcnt=0;
for(i=0; i<f;
i++)
        p[i]=9999; }
```

```
int isHit(int data)
{
h
i
t
=
0
;
    for(j=0; j<f; j++)
    {
        if(p[j]==data)

{
hit=
1;
brea
k;
} }
retu
rn
hit;
}

int getHitIndex(int data)
{     int
hitind;
for(k=0;
k<f; k++)
    {
        if(p[k]==data)
        {

hitind=k;
break;

}}
retur
n
hitin
d;
}

void dispPages()
{
    for (k=0; k<f; k++)
    {
        if(p[k]!=9999)
```

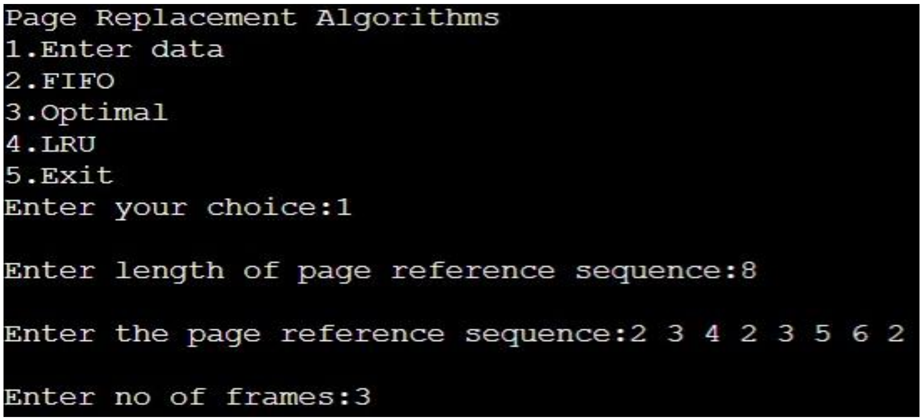
```
            printf(" %d",p[k]); }}

void dispPgFaultCnt()
    printf("\nTotal no of page faults:%d",pgfaultcnt);
v
o
i
d
f
i
f
o
(
)
{

getdata();
initialize()
;
for(i=0;
i<n; i++)
    {
        printf("\nFor %d :",in[i]);
//not a hit
if(isHit(in[i])==
0)
        {

            for(k=0; k<f-
1; k++)
p[k]=p[k+1];
p[k]=in[i];
pgfaultcnt++;
            dispPages();

}

e
l
s
e
            printf("No page fault");
    }
    dispPgFaultCnt();
}
void optimal()
```

```
{
initi
alize
();
int
near[
50];
    for(i=0; i<n; i++)
    {
        printf("\nFor %d :",in[i]);
if(isHit(in[i])==0)
        {
            for(j=0; j<f; 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();
}

vo
id
lr
u(
) {
ini
tia
liz
e(
);

    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;
```

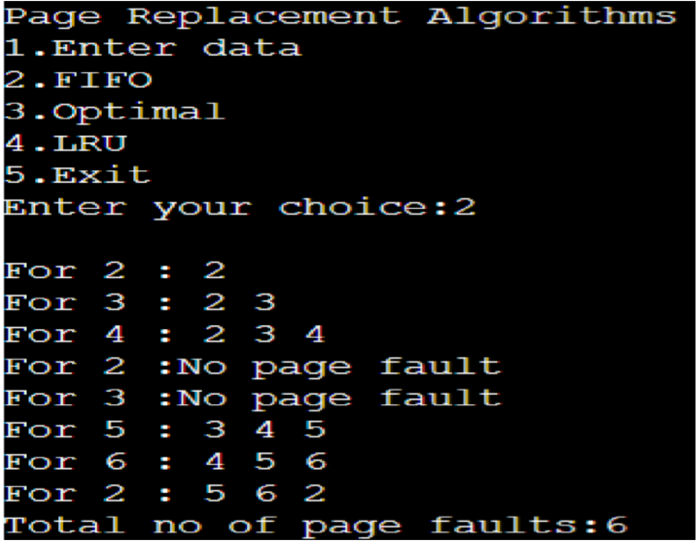
```
      }
      int
min=9999;
int repindex;
      for(j=0; j<nf; j++)
      {
        if(least[j]<min)
        {
          min=least[j];
repindex=j;
        }
      }
      p[repindex]=in[i];
      pgfaultcnt++;

      dispPages();

}

e
l
s
e
      printf("No page fault!");
  }
  dispPgFaultCnt();
}

int main()
{
int
ch
oi
ce;
  while(1)
  {
    printf("\nPage Replacement Algorithms\n
1.Enter data\n 2.FIFO\n 3.Optimal\n 4.LRU\n 5.Exit\n Enter 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;
        }

}

}
```

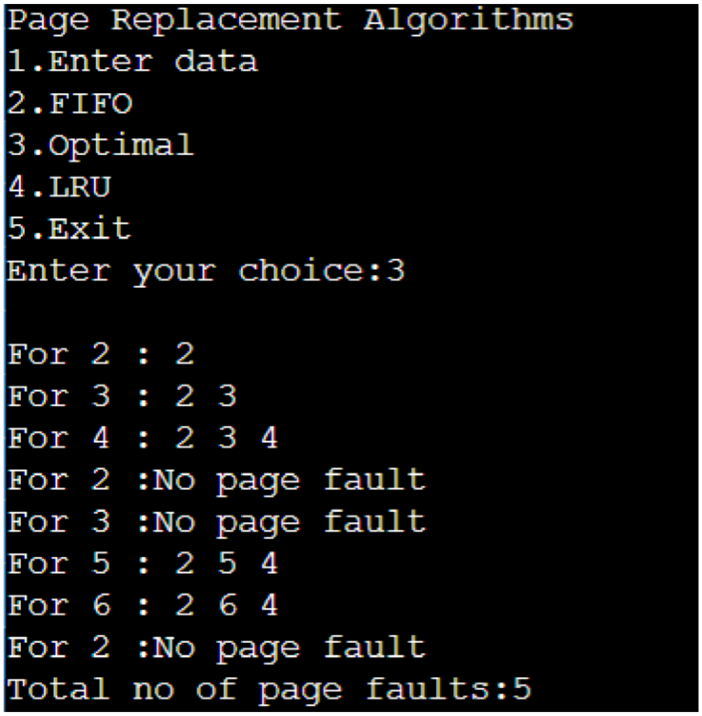
Output:



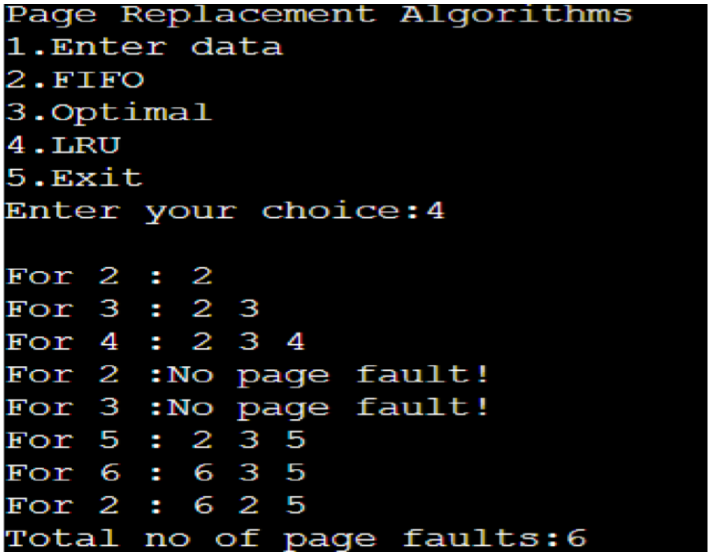
```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:1

Enter length of page reference sequence:8

Enter the page reference sequence:2 3 4 2 3 5 6 2

Enter no of frames:3
```



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

For 2 : 2
For 3 : 2 3
For 4 : 2 3 4
For 2 :No page fault
For 3 :No page fault
For 5 : 3 4 5
For 6 : 4 5 6
For 2 : 5 6 2
Total no of page faults:6
```



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

For 2 : 2
For 3 : 2 3
For 4 : 2 3 4
For 2 :No page fault
For 3 :No page fault
For 5 : 2 5 4
For 6 : 2 6 4
For 2 :No page fault
Total no of page faults:5
```



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

For 2 : 2
For 3 : 2 3
For 4 : 2 3 4
For 2 :No page fault!
For 3 :No page fault!
For 5 : 2 3 5
For 6 : 6 3 5
For 2 : 6 2 5
Total no of page faults:6
```