NAME: MEET SHAH

USN: 1PE17CS084

Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.

Page Replacement Algorithms

1. FIFO – This is the simplest page replacement algorithm. 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 page in the front of the queue is selected for removal.

2. LRU – In this algorithm page will be replaced which is least recently used.

PROGRAM CODE:

```
#include<stdio.h>
#include<stdlib.h>
void FIFO(char [ ],char [ ],int,int);
void lru(char [ ],char [ ],int,int);
void opt(char [ ],char [ ],int,int);
int main()
{
int ch,YN=1,i,l,f;
char F[10],s[25];
printf("\nEnter the no of empty frames: ");
scanf("%d",&f);
printf("\nEnter the length of the string: ");
scanf("%d",&l);
printf("\nEnter the string: ");
scanf("%s",s);
for(i=0;i<f;i++)
F[i]=-1;
do
{
printf("\n********** MENU **********");
printf("\n1:FIFO\n2:LRU \n3:EXIT");
```

USN: 1PE17CS084

```
printf("\nEnter your choice: ");
scanf("%d",&ch);
switch(ch)
{
case 1: for(i=0;i<f;i++)
F[i]=-1;
FIFO(s,F,l,f);

break;

case 2: for(i=0;i<f;i++)
F[i]=-1;
lru(s,F,l,f);
break;
case 3: exit(0);
}
printf("\n\nDo u want to continue IF YES PRESS 1\nIF NO PRESS 0 :");
scanf("%d",&YN);
} while(YN==1);
return(0);
}
//FIFO
void FIFO(char s[],char F[],int l,int f)
{
int i,j=0,k,flag=0,cnt=0;
printf("\n\tPAGE\t FRAMES\t\t\t FAULTS");
for(i=0;i<l;i++)
{
for(k=0;k<f;k++)
{
if(F[k]==s[i])
flag=1;
}
if(flag==0)
{
printf("\n\t%c\t",s[i]);
F[j]=s[i];
```

USN: 1PE17CS084

```
 j++;
 for(k=0;k<f;k++)
 printf(" %c",F[k]);
 printf("\tPage-fault%d",cnt);
 cnt++;
 }
 else
 {
 flag=0;
 printf("\n\t%c\t",s[i]);
 for(k=0;k<f;k++)
 printf(" %c",F[k]);
 printf("\tNo page-fault");
 }
 if(j==f)
 j=0;
 }
 }
 //LRU
 void lru(char s[],char F[],int l,int f)
 {
 int i,j=0,k,m,flag=0,cnt=0,top=0;
 printf("\n\tPAGE\t FRAMES\t\t\t FAULTS");
 for(i=0;i<l;i++)
 {
 for(k=0;k<f;k++)
 {
 if(F[k]==s[i])
 {
 flag=1;
 break;
 }
 }
 printf("\n\t%c\t",s[i]);
 if(j!=f && flag!=1)
 {
 F[top]=s[i];
```

```
j++;
if(j!=f)
top++;
}
else
{
if(flag!=1)
{
for(k=0;k<top;k++)
F[k]=F[k+1];
F[top]=s[i];
}
if(flag==1)
{
for(m=k;m<top;m++)
F[m]=F[m+1];
F[top]=s[i];
}
}
for(k=0;k<f;k++)
printf(" %c",F[k]);
if(flag==0)
{
printf("\tPage-fault%d",cnt);
cnt++;
}
else
printf("\tNo page fault");
flag=0;
}
}
```

USN: 1PE17CS084

OUTPUT: