Ex.No:13.a	PAGE REPLACEMENT ALGORITHMS
	FIFO

AIM:

To write a C program for implementation of FIFO page replacement algorithm.

ALGORITHM:

Step 1: Start the program.
Step 2: Declare the necessary variables.
Step 3: Enter the number of frames.
Step 4: Enter the reference string end with zero.
Step 5: FIFO page replacement selects the page that has been in memory the longest time and when the page must be replaced the oldest page is chosen.
Step 6: When a page is brought into memory, it is inserted at the tail of the queue.
Step 7: Initially all the three frames are empty.
Step 8: The page fault range increases as the no of allocated frames also increases.
Step 9: Print the total number of page faults.
Step 10: Stop the program.

PROGRAM:

```
#include<stdio.h>
int main()
{
int i=0,j=0,k=0,i1=0,m,n,rs[30],flag=1,p[30];
system("clear");
printf("FIFO page replacement algorithm. ..\\n");
printf("enter the no. of frames:");
scanf("%d",&n);
printf("enter the reference string:");
while(1)
{
scanf("%d",&rs[i]);
if(rs[i]==0)
break;
i++;
}
m=i;
for(j=0;j<n;j++)
p[j]=0;
for(i=0;i<m;i++)
```

```
{
flag=1;
for(j=0;j<n;j++)
if(p[j]==rs[i])
{
printf("data already in page... \n");
flag=0;
break;
}
if(flag==1)
{
p[i1]=rs[i];
i1++;
k++;
if(i1==n)
i1=0;
for(j=0;j<n;j++)
{
printf("\n page %d:%d",j+1,p[j]);
if(p[j]==rs[i])
printf("*");
}
printf("\n\n");
}
}
printf("total no page faults=%d",k);
}
```

OUTPUT:

Ex.No:13.b	PAGE REPLACEMENT ALGORITHMS
	LRU

AIM:

To write a c program to implement LRU page replacement algorithm.

ALGORITHM:

Step 1: Start the process
Step 2: Declare the size
Step 3: Get the number of pages to be inserted
Step 4: Get the value
Step 5: Declare counter and stack
Step 6: Select the least recently used page by counter value
Step 7: Stack them according the selection.
Step 8: Display the values
Step 9: Stop the process

ROGRAM:

```
#include<stdio.h>
main()
{
int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
printf("Enter no of pages:");
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
scanf("%d",&p[i]);
printf("Enter no of frames:");
scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)
{
c1=0;
for(j=0;j<f;j++)
{
if(p[i]!=q[j])
c1++;
```

```
}
if(c1==f)
{c++;
if(k<f)
{q[k]=p[i];
k++;
for(j=0;j<k;j++)
printf("\t%d",q[j]);
printf("\n");
}
else
{for(r=0;r<f;r++)
{c2[r]=0;
for(j=i-1;j<n;j--)
{if(q[r]!=p[j])
c2[r]++;
else
break;
}}
for(r=0;r<f;r++)
b[r]=c2[r];
for(r=0;r<f;r++)
{
for(j=r;j<f;j++)
{
if(b[r]<b[j])
{
t=b[r];
b[r]=b[j];
b[j]=t;
}}}
for(r=0;r<f;r++)
{
if(c2[r]==b[0])
q[r]=p[i];
printf("\t%d",q[r]);
}
printf("\n");
}}}
printf("\nThe no of page faults is %d",c);
}
```