

## FIFO :

```
#include<stdio.h>
int main()
{
    int i,j,n,a[50],f[10],no,k,avail,c=0;
    printf("Enter no. of pages: ");
    scanf("%d",&n);
    printf("Enter the page no.: ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("Enter number of frames: ");
    scanf("%d",&no);
    for(i=0;i<no;i++)
    {
        f[i]=-1;
        j=0;
    }
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
        avail=0;
        for(k=0;k<no;k++)
        {
            if(f[k]==a[i])
            {
                avail=1;
            }
        }
        if(avail==0)
        {
            f[j]=a[i];
            j=(j+1)%no;
            c++;
            for(k=0;k<no;k++)
            {
                printf("%d\t",f[k]);
            }
        }
        printf("\n");
    }
    printf("No. of hits: %d",n-c);
    printf("\nPage fault is: %d",c);
    return 0;
}
```

## OPTIMAL :

```
#include<stdio.h>
int main()
{
    int nof,nop,f[10],p[30],t[10],f1,f2,f3,i,j,k,pos,max,faults=0;
    printf("Enter no. of frames: ");
    scanf("%d",&nof);
    printf("Enter no. of pages: ");
    scanf("%d",&nop);
    printf("Enter the pages: ");
    for(i=0;i<nop;i++)
    {
        scanf("%d",&p[i]);
    }
    for(i=0;i<nof;i++)
    {
        f[i]=-1;
    }
    for(i=0;i<nop;i++)
    {
        f1=f2=0;
        for(j=0;j<nof;j++)
        {
            if(f[j]==p[i])
            {
                f1=f2=1;
                break;
            }
        }
    }

    if(f1==0)
    {
        for(j=0;j<nof;j++)
        {
            if(f[j]==-1)
            {
                faults++;
                f[j]=p[i];
                f2=1;
                break;
            }
        }
    }
    if(f2==0)
    {
        f3=0;
        for(j=0;j<nof;j++)
        {
            t[j]=-1;
            for(k=i+1;k<nop;k++)
```

```

        {
            if(f[j]==p[k])
            {
                t[j]=k;
                break;
            }
        }
    }
    for(j=0;j<nof;j++)
    {
        if(t[j]==-1)
        {
            pos=j;
            f3=1;
            break;
        }
    }
    if(f3==0)
    {
        max=t[0];
        pos=0;
        for(j=0;j<nof;j++)
        {
            if(t[j]>max)
            {
                max=t[j];
                pos=j;
            }
        }
    }
    f[pos]=p[i];
    faults++;
}
printf("\n");
for(j=0;j<nof;j++)
{
    printf("%d\t",f[j]);
}
}
printf("\nTotal page faults: %d",faults);
printf("\nTotal hits: %d",nop-faults);
return 0;
}

```

## LRU :

```

#include<stdio.h>
main()
{
    int q[20],p[50],c=0,cl,d,f,i,j,k=0,n,r,t,b[20],x[20];
    printf("Enter number of pages: ");
}

```

```

scanf("%d",&n);
printf("Enter the pages: ");
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=0;i<n;i++)
{
    cl=0;
    for(j=0;j<f;j++)
    {
        if(p[i]!=q[j])
        {
            cl++;
        }
    }
    if(cl==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++)
            {
                x[r]=0;
                for(j=i-1;j<n;j--)
                {
                    if(q[r]!=p[j])
                    {
                        x[r]++;
                    }
                    else
                    {
                        break;
                    }
                }
            }
        }
    }
}

```

```

    for(r=0;r<f;r++)
    {
        b[r]=x[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(x[r]==b[0])
        {
            q[r]=p[i];
        }
        printf("\t%d",q[r]);
    }
    printf("\n");
}
else
{
    printf("\n");
}
}
printf("\nNo. of page faults: %d",c);
printf("\nNo. of hits: %d",n-c);
}

```

