

## LEAST RECENTLY USED ALGORITHM

```
#include<stdio.h>

#include<conio.h>

void main()

{

int q[20],p[50],c=0,cl,d,f,i,j,k=0,n,r,t,b[20],c2[20];

clrscr();

printf("\n LRU Page Replacement Algorithm");

printf("\n *****");

printf("\n Enter the no of pages:");

scanf("%d",&n);

printf("Enter the reference string:");

for(i=0;i<n;i++)

{

scanf("%d",&p[i]);

}

printf("Enter the 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++)

{

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++)  
        {  
            c2[r]=0;  
            for(j=i-1;j<n;j--)  
            {  
                if(q[r]!=p[j]) c2[r]++;  
            }  
            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++)  
{  
j++;  
if(c2[r]==b[0])  
q[r]=p[i];  
printf("\t%d",q[r]);  
}  
printf("\n");  
}  
}  
}  
printf("\n The no of page fault is %d",c);  
getch();  
}
```

## OUTPUT:

```
LRU Page Replacement Algorithm
*****
Enter the no of pages:10
Enter the reference string:7 5 9 4 3 7 9 6 2 1
Enter the no of Frames:3

    7
    7      5
    7      5      9
    4      5      9
    4      3      9
    4      3      7
    9      3      7
    9      6      7
    9      6      2
    1      6      2

The no of page fault is 10_
```

```

#include <stdio.h>

int main() {
    int n, quantum, i, t, completed;
    int processes[10], bt[10], at[10], rem_bt[10], wt[10], tat[10];
    float total_tat, total_wt;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    printf("Enter arrival times and burst times for each process:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d Arrival Time: ", i + 1);
        scanf("%d", &at[i]);
        printf("Process %d Burst Time: ", i + 1);
        scanf("%d", &bt[i]);
        processes[i] = i + 1;
        rem_bt[i] = bt[i];
        wt[i] = 0;
        tat[i] = 0;
    }
    printf("Enter time quantum: ");
    scanf("%d", &quantum);
    t = 0;
    completed = 0;
    while (completed < n) {
        for (i = 0; i < n; i++) {

```

```
if (rem_bt[i] > 0 && at[i] <= t) {  
    if (rem_bt[i] > quantum) {  
        t += quantum;  
        rem_bt[i] -= quantum;  
    }  
    else  
    {  
        t += rem_bt[i];  
        wt[i] = t - bt[i] - at[i];  
        rem_bt[i] = 0;  
        tat[i] = t - at[i];  
        completed++;  
    }  
}  
}  
}  
}  
  
printf("\nProcess\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n");  
total_wt = 0, total_tat = 0;  
for (i = 0; i < n; i++) {  
    total_wt += wt[i];  
    total_tat += tat[i];  
    printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", processes[i], at[i], bt[i], wt[i], tat[i]);  
}  
  
printf("\nAverage waiting time: %.2f", total_wt / n);
```

```
printf("\nAverage turnaround time: %.2f\n", total_tat / n);  
return 0;  
}
```

OUTPUT:

Enter number of processes: 3

Enter arrival times and burst times for each process:

Process 1 Arrival Time: 0

Process 1 Burst Time: 8

Process 2 Arrival Time: 1

Process 2 Burst Time: 4

Process 3 Arrival Time: 2

Process 3 Burst Time: 9

Enter time quantum: 3

PROCESS	ARRIVAL TIME	BURST TIME	WAITING TIME	TURNAROUND TIME
1	0	10	10	18
2	1	8	8	12
3	2	10	10	19

Average waiting Time: 9.33

Average turnaround time: 16.33

## PRIORITY SCHEDULING ALGORITHM

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[20][20],chng[20];
int wt[20],a[20],n,i,j,temp,trn[20],p[20];
float tot,t;
clrscr();
printf("\nPriority scheduling algorithm");
printf("\n*****");
printf("\nEnter the no.of process:");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
printf("Enter process id and burst time and priority\n");
scanf("%s%d%d",&s[i],&a[i],&p[i]);
}
wt[0]=0;
a[0]=0;
t=tot=0;
for(i=1;i<=n;i++)
{
for(j=i+1;j<=n;j++)
{
if(p[i]>p[j])
```



```

{
temp=a[i];
a[i]=a[j];
a[j]=temp;
temp=p[i];
p[i]=p[j];
p[j]=temp;
strcpy(chng,s[i]);
strcpy(s[i],s[j]);
strcpy(s[j],chng);
}
}
}
printf("\n-----");
printf("\nProcess Burst Time Waiting Time Turnaround Time Priority");
printf("\n-----");
for(i=1;i<=n;i++)
{
wt[i]=wt[i-1]+a[i-1];
trn[i]=wt[i]+a[i];
printf("\n%s\t%d\t\t%d\t\t%d\t\t%d",s[i],a[i],wt[i],trn[i],p[i]);
tot=tot+wt[i];
t=t+trn[i];
}
printf("\n-----");
printf("\nAverage Waiting time=%6.2f",tot/n);
printf("\nAverage Turnaround Time=%6.2f",t/n);

```

```
getch();  
}
```

OUTPUT:

Priority Scheduling Algorithm

\*\*\*\*\*

Enter the no.of process: 2

Enter the process id and burst time and priority

P1 65 26

Enter the process id and burst time and priority

P2 55 87

Process	Burst Time	Waiting Time	Turnaround Time	Priority
P1	65	0	65	26
P2	55	65	120	87

-----  
Average Waiting time=32.50

Average Turnaround time= 92.50