Prn no:2262701242059

```
#include<stdio.h>
int findLRU(int time[], int n){
int i, minimum = time[0], pos = 0;
for(i = 1; i < n; ++i){
if(time[i] < minimum){
minimum = time[i];
pos = i;
}}
return pos;
}
int main()
{
int no_of_frames, no_of_pages, frames[10], pages[30], counter = 0,
time[10], flag1, flag2, i, j, pos, faults = 0;
printf("Enter number of frames: ");
scanf("%d", &no_of_frames);
printf("Enter number of pages: ");
scanf("%d", &no_of_pages);
printf("Enter reference string: ");
for(i = 0; i < no_of_pages; ++i){
scanf("%d", &pages[i]);
}
for(i = 0; i < no_of_frames; ++i){
frames[i] = -1;
}
for(i = 0; i < no_of_pages; ++i){
flag1 = flag2 = 0;
for(j = 0; j < no_of_frames; ++j){
        if(frames[j] == pages[i]){
counter++;
```

```
time[j] = counter;
flag1 = flag2 = 1;
break;
}
}
if(flag1 == 0){
for(j = 0; j < no_of_frames; ++j){
if(frames[j] == -1){
counter++;
faults++;
frames[j] = pages[i];
time[j] = counter;
flag2 = 1;
break;
}
}
}
if(flag2 == 0){
pos = findLRU(time, no_of_frames);
counter++;
faults++;
frames[pos] = pages[i];
time[pos] = counter;
}
printf("\n");
for(j = 0; j < no_of_frames; ++j){
printf("%d\t", frames[j]);
}
}
printf("\n\nTotal Page Faults = %d", faults);
return 0;
```

}

Output:-

```
C:\Users\soham\OneDrive\De

Enter number of frames: 2
Enter number of pages: 5
Enter reference string: 2 3 1 4 2

2       -1
2       3
1       3
1       4
2       4

Total Page Faults = 5
--------------------------------
Process exited after 16.15 seconds with return value 0
Press any key to continue . . .
```

Prn no:2262701242059

```
//C program for Banker's Algorithm
#include <stdio.h>
int main()
{
// P0, P1, P2, P3, P4 are the names of Process
int n, r, i, j, k;
n = 5; // Indicates the Number of processes
r = 3; //Indicates the Number of resources
int alloc[5][3] = { { 0, 0, 1 }, // P0 // This is Allocation Matrix
{ 3, 0, 0 }, // P1
{ 1, 0, 1 }, // P2
{ 2, 3, 2 }, // P3
{ 0, 0, 3 } }; // P4
int max[5][3] = { { 7, 6, 3 }, // P0 // MAX Matrix
{ 3, 2, 2 }, // P1
{ 8, 0, 2 }, // P2
{ 2, 1, 2 }, // P3
{ 5, 2, 3 } }; // P4
int avail[3] = { 2, 3, 2 }; // These are Available Resources
int f[n], ans[n], ind = 0;
for (k = 0; k < n; k++) {
f[k] = 0;
}
int need[n][r];
for (i = 0; i < n; i++) {
for (j = 0; j < r; j++)
```

```
need[i][j] = max[i][j] - alloc[i][j];
}
int y = 0;
for (k = 0; k < 5; k++) {
for (i = 0; i < n; i++) {
if (f[i] == 0) {
int flag = 0;
for (j = 0; j < r; j++) {
if (need[i][j] > avail[j]){
flag = 1;
break;
}
}
if (flag == 0) {
ans[ind++] = i;
for (y = 0; y < r; y++)
avail[y] += alloc[i][y];
f[i] = 1;
}
}
}
}
printf("Th SAFE Sequence is as follows\n");
for (i = 0; i < n - 1; i++)
printf(" P%d ->", ans[i]);
printf(" P%d", ans[n - 1]);
return (0);
}
```

Output:-

```
C:\Users\soham\OneDrive\De

Th SAFE Sequence is as follows
 P1 -> P3 -> P4 -> P0 -> P2
--------------------------------
Process exited after 2.026 seconds with return value 0
Press any key to continue . . .
```

Prn no:2262701242059

```
#include<stdio.h>
int main()
{
int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1, flag2,
flag3, i, j, k, pos, max, faults = 0;
printf("Enter number of frames: ");
scanf("%d", &no_of_frames);
printf("Enter number of pages: ");
scanf("%d", &no_of_pages);
printf("Enter page reference string: ");
for(i = 0; i < no_of_pages; ++i){
scanf("%d", &pages[i]);
}
for(i = 0; i < no_of_frames; ++i){
frames[i] = -1;
}
for(i = 0; i < no_of_pages; ++i){
flag1 = flag2 = 0;
for(j = 0; j < no_of_frames; ++j){
if(frames[j] == pages[i]){
flag1 = flag2 = 1;
break;
}
}
if(flag1 == 0){
for(j = 0; j < no_of_frames; ++j){
```

```
if(frames[j] == -1){
faults++;
frames[j] = pages[i];
flag2 = 1;
break;
}
}
}
if(flag2 == 0){
flag3 =0;
for(j = 0; j < no_of_frames; ++j){
temp[j] = -1;
for(k = i + 1; k < no_of_pages; ++k){
if(frames[j] == pages[k]){
temp[j] = k;
break;
}
}
}
for(j = 0; j < no_of_frames; ++j){
if(temp[j] == -1){
pos = j;
flag3 = 1;
break;
}
}
if(flag3 ==0){
max = temp[0];
pos = 0;
for(j = 1; j < no_of_frames; ++j){
if(temp[j] > max){
```

```
max = temp[j];
pos = j;
}
}
}
frames[pos] = pages[i];
faults++;
}
printf("\n");
for(j = 0; j < no_of_frames; ++j){
printf("%d\t", frames[j]);
}
}
printf("\n\nTotal Page Faults = %d", faults);
return 0;
}
```

Output:-

```
C:\Users\soham\OneDrive\De

Enter number of frames: 2
Enter number of pages: 5
Enter page reference string: 2 3 1 4 2

2       -1
2       3
2       1
2       4
2       4

Total Page Faults = 4
--------------------------------
Process exited after 10.77 seconds with return value 0
Press any key to continue . . .
```