Practical: 13

Set 1: How to take no. of processes, no. of resources, maximum resource matrix, allocated resource matrix and available resources for each process.

Code:

```
#include <stdio.h> #include <stdlib.h> int main()
{
int Max[10][10], need[10][10], alloc[10][10], avail[10], completed[10]; int p, r, i, j, process, count;
count = 0;
printf("Enter the no of processes : ");
scanf("%d", &p); for(i = 0; i< p; i++)
completed[i] = 0;
printf("\n\nEnter the no of resources : ");
scanf("%d", &r);
printf("\n\nEnter the Max Matrix for each process : "); for(i = 0; i < p; i++)
   { printf("\nFor process %d : ", i + 1);
for(j = 0; j < r; j++) scanf("%d", &Max[i][j]);
   } printf("\n\nEnter the allocation for each process : "); for(i = 0; i < p; i++)
   { printf("\nFor process %d : ",i + 1);
for(j = 0; j < r; j++) scanf("%d", &alloc[i][j]);
   } printf("\n\nEnter the Available Resources : ");
for(i = 0; i < r; i++)
scanf("%d", &avail[i]);

for(i = 0; i < p; i++)
{
for( j = 0; j < r; j++) printf("%d ", Max[i][j]);
printf("\t\t"); for( j = 0; j < r; j++) printf("%d ",
alloc[i][j]); printf("\n");
}

}
```

Set 3: Perform the Banker's Algorithm and find out that whether the System is Safe or not and also print the Safe Sequence.

Safety Algorithm:

1. Let Work and Finish be vectors of length m and n, respectively. Initially,

   Work = Available
   Finish[i] =false for i = 0, 1, ... , n - 1.

This means, initially, no process has finished and the number of available resources is represented by the Available array.

2. Find an index i such that both

   Finish[i] ==false
   Needi <= Work

   If there is no such i present, then proceed to step 4.

   It means, we need to find an unfinished process whose need can be satisfied by the available resources. If no such process exists, just go to step 4.

3. Perform the following:

   Work = Work + Allocation;
   Finish[i] = true; Go to step 2.

   When an unfinished process is found, then the resources are allocated and the process is marked finished. And then, the loop is repeated to check the same for all other processes.

4. If Finish[i] == true for all i, then the system is in a safe state.

   That means if all processes are finished, then the system is in safe state.

```
Enter the number of resources : 3

Enter the max instances of each resource
a= 10
b= 5
c= 7

 Enter the number of processes: 5

 Enter the allocation matrix
      a b c
P[0]  0 1 0
P[1]  2 0 0
P[2]  3 0 2
P[3]  2 1 1
P[4]  0 0 2

Enter the MAX matrix
      a b c
P[0]  7 5 3
P[1]  3 2 2
P[2]  9 0 2
P[3]  4 2 2
P[4]  5 3 3

        < P[1]  P[3]  P[4]  P[0]  P[2] >
```