EXPERIMENT NO. 9

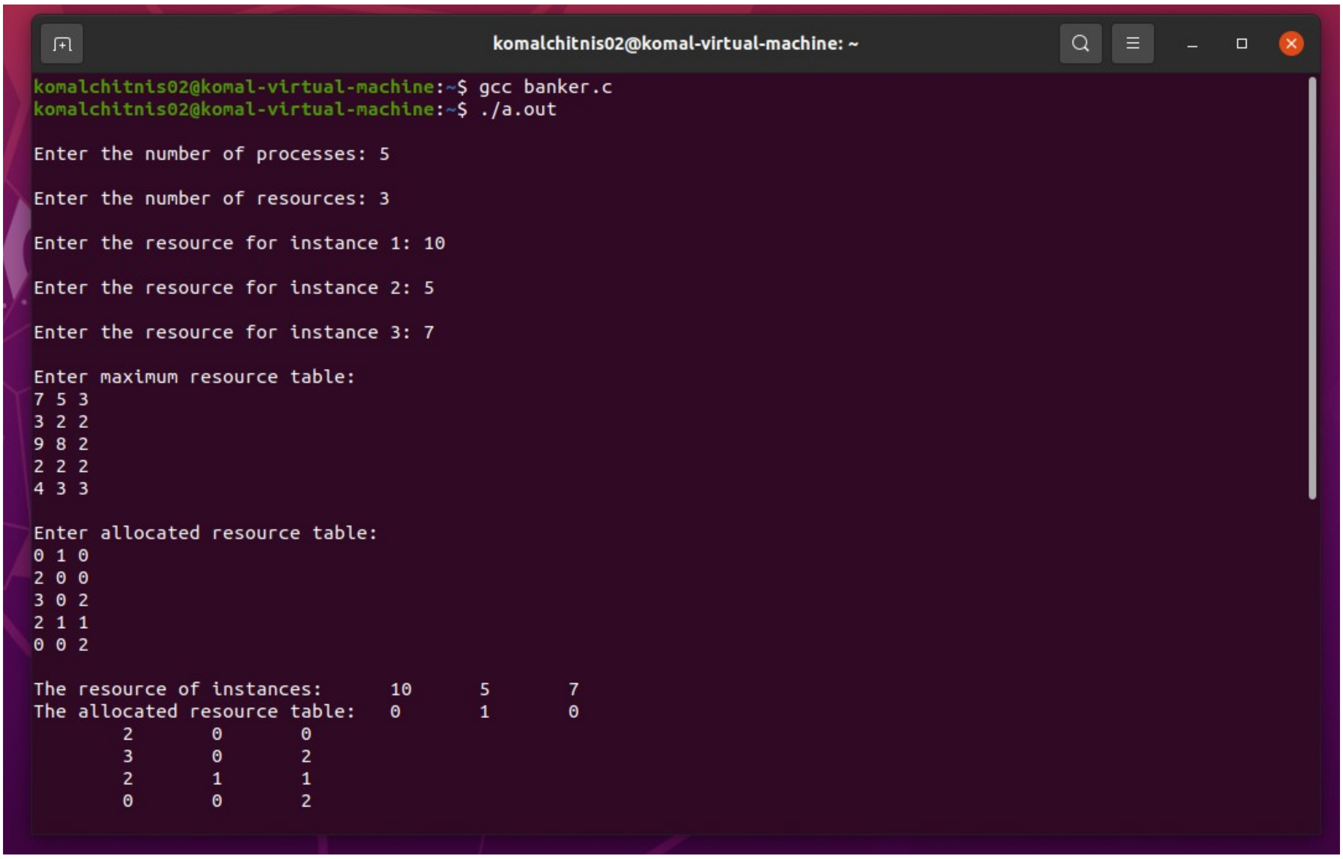
NAME: KOMAL MAHESH CHITNIS
MOODLE ID: 20102068
ROLL NO: 26
DIV: A

PROGRAM:

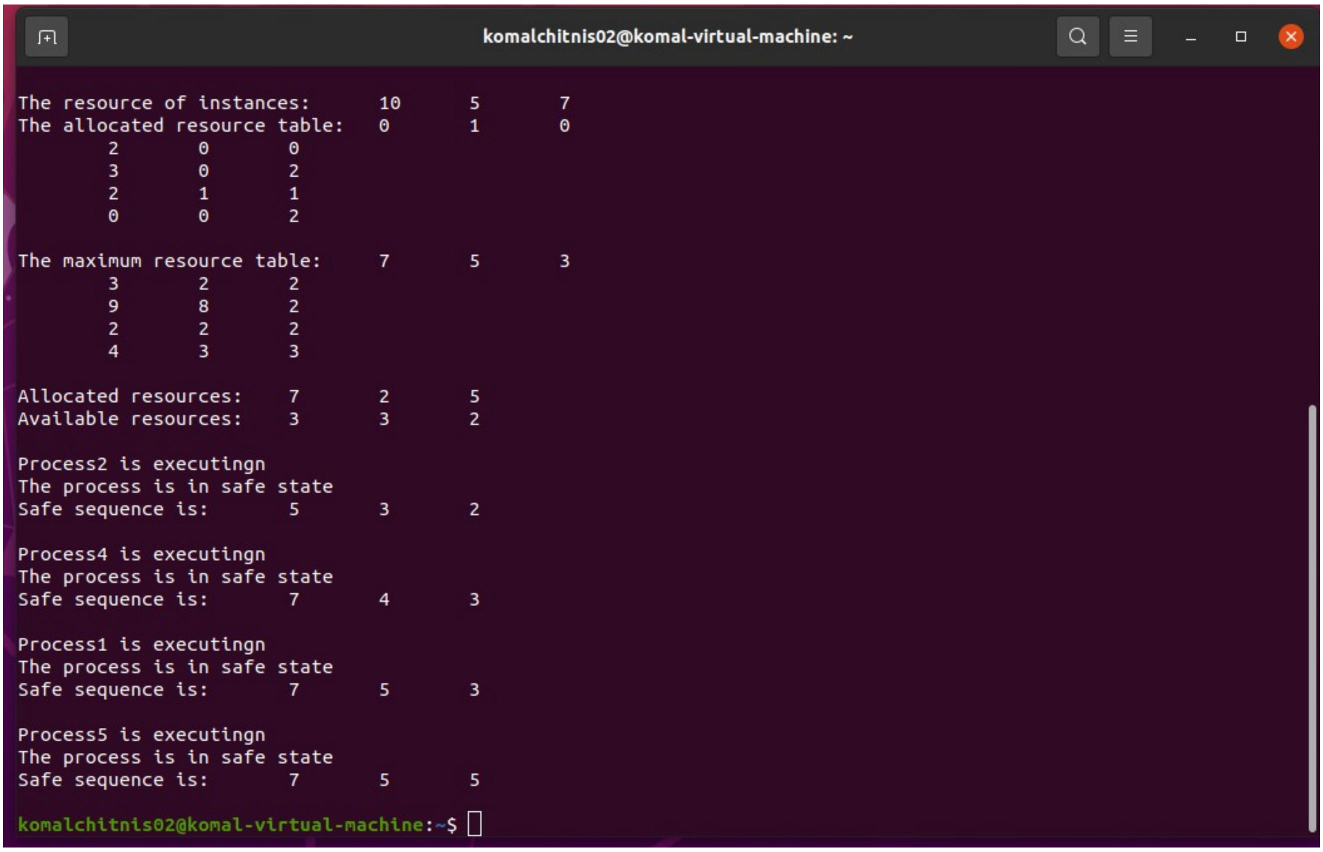
```
#include <stdio.h>
int curr[5][5], maxclaim[5][5], avl[5];
int alloc[5] = {0, 0, 0, 0, 0};
int maxres[5], running[5], safe=0;
int count = 0, i, j, exec, r, p, k = 1;
int main(){
printf("\nEnter the number of processes: ");
scanf("%d", &p);
for (i = 0; i < p; i++) {
running[i] = 1;
count++;
}
printf("\nEnter the number of resources: ");
scanf("%d", &r);
for (i = 0; i < r; i++) {
printf("\nEnter the resource for instance %d: ", k++);
scanf("%d", &maxres[i]);
}
printf("\nEnter maximum resource table:");
for (i = 0; i < p; i++) {
for(j = 0; j < r; j++) {
scanf("%d", &maxclaim[i][j]);
}
}
printf("\nEnter allocated resource table:");
for (i = 0; i < p; i++) {
for(j = 0; j < r; j++) {
scanf("%d", &curr[i][j]);
}
}
printf("\nThe resource of instances: ");
for (i = 0; i < r; i++) {
printf("\t%d", maxres[i]);
}
printf("\nThe allocated resource table:");
for (i = 0; i < p; i++) {
for (j = 0; j < r; j++) {
printf("\t%d", curr[i][j]);
```

```
}
printf("\n");
}
printf("\nThe maximum resource table:");
for (i = 0; i < p; i++) {
for (j = 0; j < r; j++) {printf("\t%d", maxclaim[i][j]);
}
printf("\n");
}

for (i = 0; i < p; i++) {
for (j = 0; j < r; j++) {
alloc[j] += curr[i][j];
}
}
printf("\nAllocated resources:");
for (i = 0; i < r; i++) {
printf("\t%d", alloc[i]);
}
for (i = 0; i < r; i++) {
avl[i] = maxres[i] - alloc[i];
}
printf("\nAvailable resources:");
for (i = 0; i < r; i++) {
printf("\t%d", avl[i]);
}
printf("\n");
//Main procedure goes below to check for unsafe
while (count != 0) {
safe = 0;
for (i = 0; i < p; i++) {
if (running[i]) {
exec = 1;
for (j = 0; j < r; j++) {
if (maxclaim[i][j] - curr[i][j] > avl[j]) {
exec = 0;
break;
}
}
if (exec) {
printf("\nProcess%d is executingn", i + 1);
running[i] = 0;
count--;
safe = 1;
for (j = 0; j < r; j++) {
avl[j] += curr[i][j];
}
break;
}
}
}
if (!safe) {
printf("\nThe processes are in unsafe state.");
break;
} else {
printf("\nThe process is in safe state");
printf("\nSafe sequence is:");
for (i = 0; i < r; i++) {
printf("\t%d", avl[i]);
}
printf("\n");
}

}
}
```

OUTPUT:



```
komalchitnis02@komal-virtual-machine:~$ gcc banker.c
komalchitnis02@komal-virtual-machine:~$ ./a.out

Enter the number of processes: 5

Enter the number of resources: 3

Enter the resource for instance 1: 10

Enter the resource for instance 2: 5

Enter the resource for instance 3: 7

Enter maximum resource table:
7 5 3
3 2 2
9 8 2
2 2 2
4 3 3

Enter allocated resource table:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2

The resource of instances:      10      5       7
The allocated resource table:   0       1       0
        2       0       0
        3       0       2
        2       1       1
        0       0       2
```



```
The resource of instances:      10      5       7
The allocated resource table:   0       1       0
        2       0       0
        3       0       2
        2       1       1
        0       0       2

The maximum resource table:     7       5       3
        3       2       2
        9       8       2
        2       2       2
        4       3       3

Allocated resources:    7       2       5
Available resources:    3       3       2

Process2 is executingn
The process is in safe state
Safe sequence is:       5       3       2

Process4 is executingn
The process is in safe state
Safe sequence is:       7       4       3

Process1 is executingn
The process is in safe state
Safe sequence is:       7       5       3

Process5 is executingn
The process is in safe state
Safe sequence is:       7       5       5

komalchitnis02@komal-virtual-machine:~$ 
```