ASSIGNMENT NO. 5

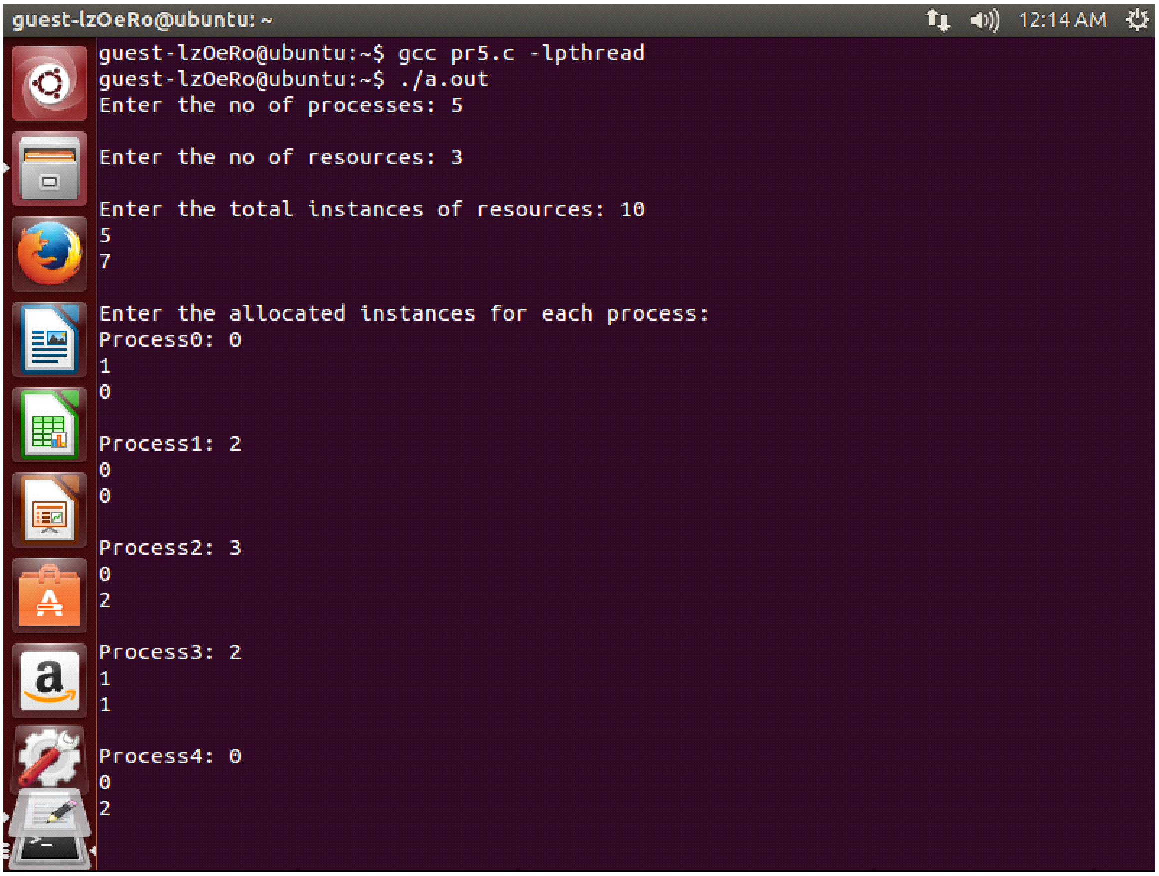
```
#include <stdio.h>
void main()
{
  int alloc[10][10], max[10][10], avail[10], tot[10], need[10][10], pflag[10] = {0},
safe[10], flag1, flag2, p, r, i, j, k = 0, m;
  printf("Enter the no of processes: ");
  scanf("%d", &p);
  printf("\nEnter the no of resources: ");
  scanf("%d", &r);
  printf("\nEnter the total instances of resources: ");
  for (i = 0; i < r; i++)
  {
  scanf("%d", &tot[i]);
  avail[i] = tot[i];
  }
  printf("\nEnter the allocated instances for each process: ");
  for (i = 0; i < p; i++)
  {
  printf("\nProcess%d: ", i);
  for (j = 0; j < r; j++)
  scanf("%d", &alloc[i][j]);
  }
  printf("\nEnter the max instances required for each process");
  for (i = 0; i < p; i++)
```

```
{

printf("\nProcess %d: ", i);

for (j = 0; j < r; j++)

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

}

printf("\nThe available matrix is: ");

for (j = 0; j < r; j++)

{

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

avail[j] = avail[j] - alloc[i][j];

printf("\t%d", avail[j]);

}

printf("\n\nThe need matrix is: ");

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

{

printf("\nProcess %d:", i);

for (j = 0; j < r; j++)

{

need[i][j] = max[i][j] - alloc[i][j];

printf("\t%d", need[i][j]);

}

}

for (m = 0; m < p; m++)

{

for (i = 0; i < p; i++)
```

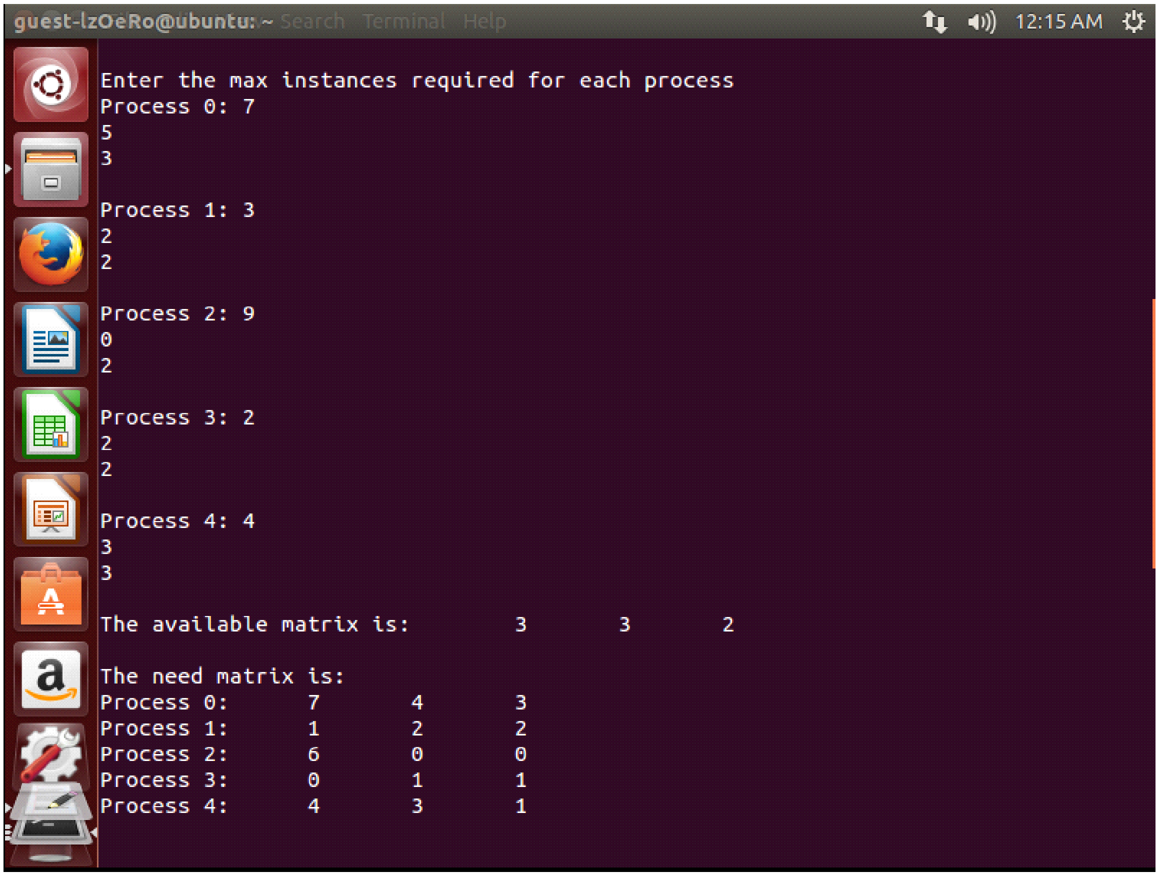
```
{
if (pflag[i] == 0)
{
flag1 = 0;
printf("\n\nFor process %d:", i);
for (j = 0; j < r; j++)
{
if (need[i][j] > avail[j])
{
flag1 = 1;
break;
}
}
if (flag1 == 0)
{
for (j = 0; j < r; j++)
avail[j] = avail[j] + alloc[i][j];
pflag[i] = 1;
printf("\nProcess %d can be granted resources..", i);
printf("\nNew Available resources are\n");
for (j = 0; j < r; j++)
printf("\t%d", avail[j]);
safe[k] = i;
k++;
}
```

```
  if (flag1 == 1)
  printf("\nProcess %d cannot be granted resources....Going to next process", i);
  } //outer if
  } //outer for
  } //outer for
  flag2 = 0;
  for (i = 0; i < p; i++)
  {
  if (pflag[i] == 0)
  {
  printf("\n\nSystem is NOT in a safe state");
  flag2 = 0;
  break;
  }
  else
  flag2 = 1;
  }
  if (flag2 == 1)
  {
  printf("\n\nSystem is in a SAFE STATE\nSAFE SEQUENCE is\n");
  for (i = 0; i < p; i++)
  printf("Process%d ", safe[i]);
  }
} //main
```

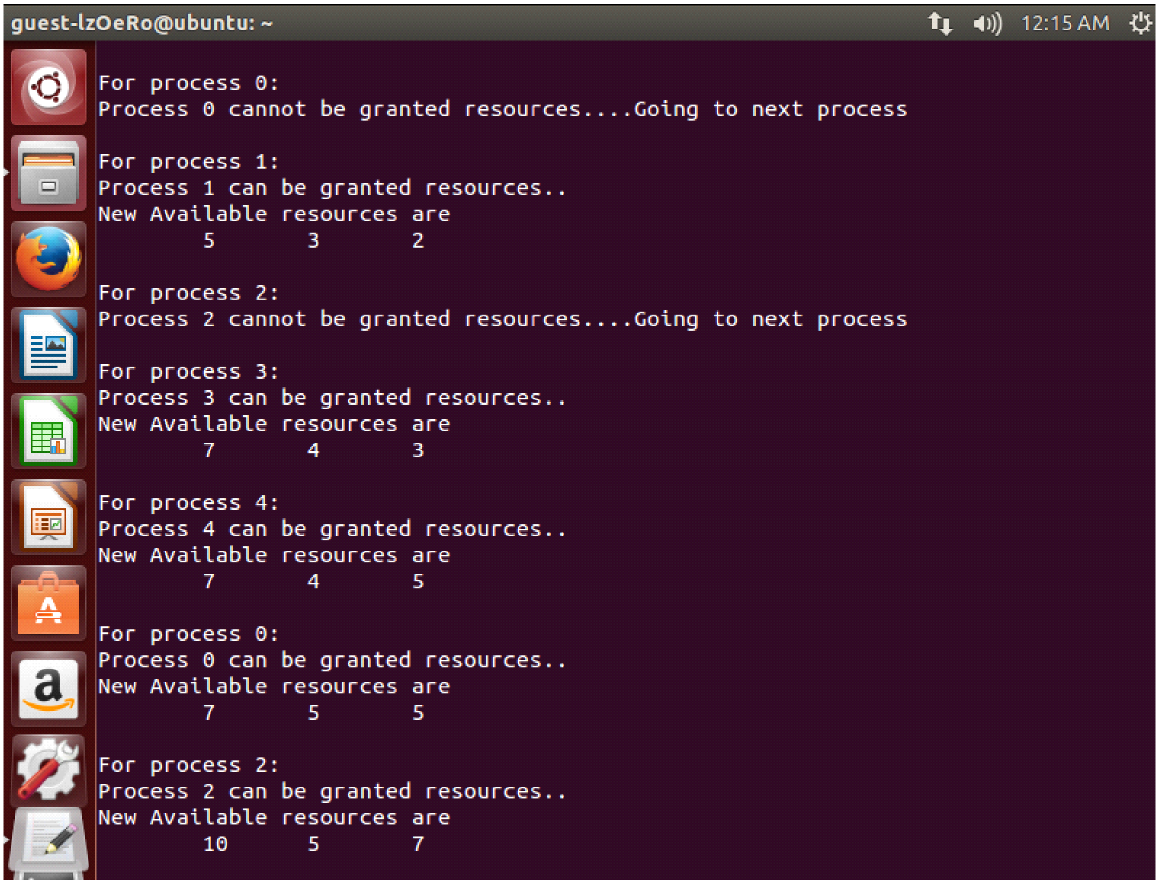
OUTPUT:



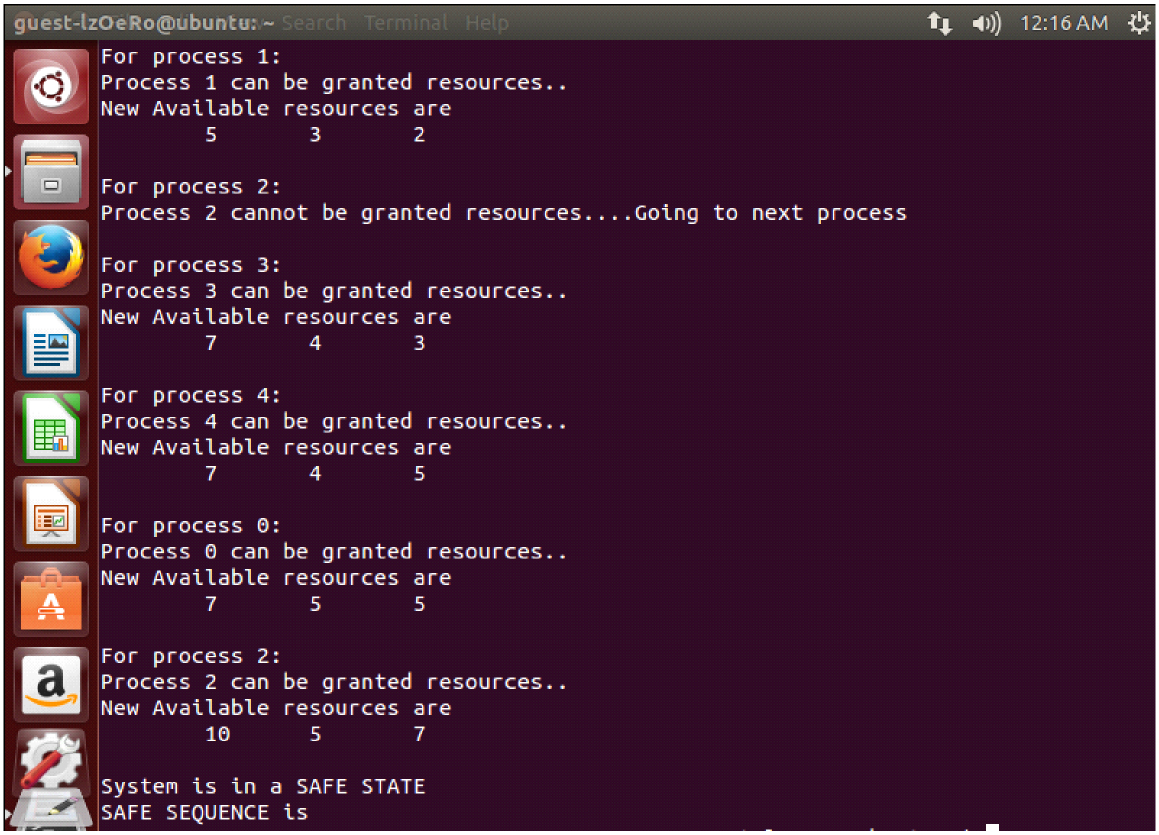
```
guest-lzOeRo@ubuntu:~$ gcc pr5.c -lpthread
guest-lzOeRo@ubuntu:~$ ./a.out
Enter the no of processes: 5

Enter the no of resources: 3

Enter the total instances of resources: 10
5
7

Enter the allocated instances for each process:
Process0: 0
1
0

Process1: 2
0
0

Process2: 3
0
2

Process3: 2
1
1

Process4: 0
0
2
```



```
Enter the max instances required for each process
Process 0: 7
5
3

Process 1: 3
2
2

Process 2: 9
0
2

Process 3: 2
2
2

Process 4: 4
3
3

The available matrix is:        3       3       2

The need matrix is:
Process 0:      7       4       3
Process 1:      1       2       2
Process 2:      6       0       0
Process 3:      0       1       1
Process 4:      4       3       1
```



```
For process 0:
Process 0 cannot be granted resources....Going to next process

For process 1:
Process 1 can be granted resources..
New Available resources are
        5       3       2

For process 2:
Process 2 cannot be granted resources....Going to next process

For process 3:
Process 3 can be granted resources..
New Available resources are
        7       4       3

For process 4:
Process 4 can be granted resources..
New Available resources are
        7       4       5

For process 0:
Process 0 can be granted resources..
New Available resources are
        7       5       5

For process 2:
Process 2 can be granted resources..
New Available resources are
        10      5       7
```



```
For process 1:
Process 1 can be granted resources..
New Available resources are
        5       3       2

For process 2:
Process 2 cannot be granted resources....Going to next process

For process 3:
Process 3 can be granted resources..
New Available resources are
        7       4       3

For process 4:
Process 4 can be granted resources..
New Available resources are
        7       4       5

For process 0:
Process 0 can be granted resources..
New Available resources are
        7       5       5

For process 2:
Process 2 can be granted resources..
New Available resources are
        10      5       7

System is in a SAFE STATE
SAFE SEQUENCE is
```