

Banker:

The image shows two screenshots of a VMware Workstation terminal window. The terminal is running a program named 'banker' which simulates a resource allocation system. The output shows a series of requests and releases from different customers, each with a binary string representing a resource state and a status indicating whether the request was accepted or not.

Screenshot 1 (Top): The terminal shows the command `make banker` and the output of the program. The output includes the following lines:

```
amir47@ubuntu:~/Desktop/7$ make banker
amir47@ubuntu:~/Desktop/7$ ./banker.out 4 6 6 5 4
Customer #2 requesting (3 2 0 2 1) ==> request not accepted
Customer #3 requesting (0 1 2 3 0) ==> request not accepted
Customer #3 releasing (0 0 0 0 0)
Customer #3 requesting (1 1 1 2 0) ==> request not accepted
Customer #3 releasing (0 0 0 0 0)
Customer #3 releasing (0 0 0 0 0)
Customer #3 releasing (0 0 0 0 0)
Customer #3 requesting (1 0 1 1 1) ==> request not accepted
Customer #3 releasing (0 0 0 0 0)
Customer #3 requesting (0 3 1 0 2) ==> request not accepted
Customer #1 requesting (1 0 1 2 0) ==> request accepted
Customer #1 releasing (0 0 0 0 0)
Customer #1 requesting (0 0 1 1 0) ==> request not accepted
Customer #1 requesting (0 1 0 1 0) ==> request not accepted
Customer #1 releasing (0 0 1 2 0)
Customer #1 releasing (0 0 0 0 0)
Customer #1 releasing (0 0 0 0 0)
Customer #1 releasing (1 0 0 0 0)
Customer #1 releasing (0 0 0 0 0)
Customer #1 requesting (0 1 0 1 2) ==> request accepted
Customer #4 requesting (0 2 0 2 0) ==> request accepted
Customer #4 requesting (1 0 1 1 0) ==> request accepted
Customer #4 releasing (0 0 0 0 0) ==> request accepted
Customer #4 releasing (1 1 0 1 0)
Customer #4 requesting (1 1 0 2 1) ==> request accepted
Customer #4 releasing (1 1 1 2 1)
Customer #4 requesting (0 0 1 1 1) ==> request accepted
Customer #4 requesting (1 0 0 0 0) ==> request accepted
Customer #4 requesting (0 0 0 1 0) ==> request accepted
Customer #4 requesting (0 0 0 0 0) ==> request accepted
Customer #2 requesting (0 0 2 0 2) ==> request not accepted
Customer #2 releasing (0 0 0 0 0)
Customer #2 requesting (3 3 2 1 1) ==> request not accepted
Customer #2 releasing (0 0 0 0 0)
Customer #2 requesting (4 3 2 2 2) ==> request not accepted
Customer #2 releasing (0 0 0 0 0)
```

Screenshot 2 (Bottom): The terminal shows the continuation of the program's output. The output includes the following lines:

```
Customer #1 requesting (0 0 1 1 0) ==> request not accepted
Customer #1 requesting (0 1 0 1 0) ==> request not accepted
Customer #1 releasing (0 0 1 2 0)
Customer #1 releasing (0 0 0 0 0)
Customer #1 releasing (0 0 0 0 0)
Customer #1 releasing (1 0 0 0 0)
Customer #1 releasing (0 0 0 0 0)
Customer #1 requesting (0 1 0 1 2) ==> request accepted
Customer #4 requesting (0 2 0 2 0) ==> request accepted
Customer #4 requesting (1 0 1 1 0) ==> request accepted
Customer #4 releasing (0 0 0 0 0) ==> request accepted
Customer #4 releasing (1 1 0 2 1)
Customer #4 requesting (0 0 1 1 1) ==> request accepted
Customer #4 requesting (1 0 0 0 0) ==> request accepted
Customer #4 requesting (0 0 0 1 0) ==> request accepted
Customer #4 requesting (0 0 0 0 0) ==> request accepted
Customer #2 requesting (0 0 2 0 2) ==> request not accepted
Customer #2 releasing (0 0 0 0 0)
Customer #2 requesting (3 3 2 1 1) ==> request not accepted
Customer #2 releasing (0 0 0 0 0)
Customer #2 requesting (1 2 1 0 2) ==> request not accepted
Customer #2 requesting (4 2 0 1 0) ==> request not accepted
Customer #5 releasing (0 0 0 0 0)
Customer #5 requesting (2 1 0 0 1) ==> request accepted
Customer #5 releasing (0 1 0 0 0)
Customer #5 releasing (0 0 0 0 1)
Customer #5 releasing (0 0 0 0 0)
Customer #5 requesting (0 0 3 1 3) ==> request not accepted
Customer #5 requesting (1 0 2 2 2) ==> request not accepted
Customer #5 releasing (0 0 0 0 0)
Customer #5 requesting (1 1 2 2 0) ==> request not accepted
Customer #5 requesting (0 1 1 1 0) ==> request accepted
amir47@ubuntu:~/Desktop/7$
```