Problem Statement-

Consider a plant, where there are 10 part workers whose jobs are to produce three types of parts (A, B,C). Each of them produces three pieces of parts each time, such as (1,1,1),(3,0,0), (1,2,0), (0,1,2), (0,0,3) etc. All possible combinations will carry the same probability. In your program you have to randomly generate one combination. Each part worker will attempt to place the parts generated to a buffer area. Therefore, each combination (a, b, c) from a part worker is referred to as a place request.

In addition, there are 7 product workers whose jobs are to take the parts from the buffer area and assemble them into products. Each of them needs four pieces of parts each time; however, the four pieces will be from exactly two types of parts, such as (3,1,0), (2,2,0), (0,2,2), (0,1,3), etc. with equal occurrence probability. For example, a product worker will not make a request of (1,1,2), (4,0,0), etc.

Each such combination from a product worker is referred to as a pickup request. In your program, you need to randomly generate one combination. Moreover, a buffer area to hold parts for part workers has a capacity of 6 type A parts, 5 type B parts, and 4 type C parts. At any moment, the numbers of parts of each type (a, b, c) is referred to as buffer state. A part work can place parts to the buffer area up to the capacity, and wait for available space for remaining part(s) in the place request if needed. For example, if the current buffer state is (5, 4, 4) and a part worker generates a (1,1,1) place request, then the buffer state will be updated to (6, 5,4) and the part worker's place request will be updated to (0,0,1) and the part worker has to wait until the space for type C becomes available. Similarly, a product work can pick up those available parts from the buffer area and wait for the remaining part(s). For example, if the current buffer state is (2,1, 0) and a product worker generates a pickup request of (2,0,2), then the buffer state becomes (0,1,0) and the product worker has to wait with a updated pickup request of
(0,0,2).

In your simulation code, you should intend to allow each PartWorker and each ProductWork to complete 7 requests. Sometimes, you might not be able to accomplish the goal. That is, sometimes your program can encounter a deadlock and has to be stopped. During grading, execution sequences will be examined for correctness; rather than a guaranteed execution completion.

When a ParkWorker generates a place request or is awaken to continue a place-request, you need to print ID, the current buffer state and the place request, and the updated buffer state and updated placerequest. If the buffer have sufficient space for the place request, then the updated place request will be (0,0,0). Similarly, when a ProductWorker generates a pickup request or is awaken to continue a pickup request, you need to print the current buffer state and the pickup request, and the updated buffer state and updated pickup request. If the buffer have all the requested parts for the pickup request, then the
updated place-request will be (0,0,0).

The following are examples for a place request and a pickup request.
PartWorker ID: 8
Buffer State: (5,2,3)
Place Request: (2,0,1)
Updated Buffer State: (6,2,4)
Updated Place Request: (1,0,0)

ProductWorker ID: 5
Buffer State: (2,3,4)
Pickup Request: (3,1,0)
Updated Buffer State: (0,2,4)
Updated Pickup Request: (1,0,0)

Note-
Depending on numbers generated randomly, deadlock may occur.