Please make sure that your code will run under either Python 2.6.6 or Python 3.5.1.  
Please post your code and output to a repository on github or bitbucket and share the repository.  
  
  
------------------------------------------------------  
  
The object of the exercise is to show off your coding style and skill.  
  
Initial conditions:  
  Initially, the system contains inventory of  
  A x 150  
  B x 150  
  C x 100  
  D x 100  
  E x 200  
  
  Initially, the system contains no orders  
  
Data source:  
  There should be a data source capable of generating one or more streams of orders.  
  An order consists of a unique identifier (per stream) we will call the "header", and a demand for between zero and five units each of A,B,C,D, and E, except that there must be at least one unit demanded.  
  A valid order (in whatever format you choose): {"Header": 1, "Lines": {"Product": "A", "Quantity": "1"},{"Product": "C", "Quantity": "4"}}  
  An invalid order: {"Header": 1, "Lines": {"Product": "B", "Quantity": "0"}}  
  Another invalid order: {"Header": 1, "Lines": {"Product": "D", "Quantity": "6"}}  
  Feel free to identify streams as you'd like.  
  
Inventory allocator:  
  There should be an inventory allocator which allocates inventory to the inbound data according to the following rules:  
  1) Inbound orders to the allocator should be individually identifyable (ie two streams may generate orders with an identical header, but these orders should be identifyable from their streams)  
  2) Inventory should be allocated on a first come, first served basis; once allocated, inventory is not available to any other order.  
  3) Inventory should never drop below 0.  
  4) If a line cannot be satisfied, it should not be allocated.  Rather, it should be  backordered (but other lines on the same order may still be satisfied).  
  5) When all inventory is zero, the system should halt and produce output listing, in the order received by the system, the header of each order, the quantity on each line, the quantity allocated to each line, and the quantity backordered for each line.  
  For instance:  
  If the initial conditions are:  
  A x 2  
  B x 3  
  C x 1  
  D x 0  
  E x 0  
  
  And the input is:  
  {"Header": 1, "Lines": {"Product": "A", "Quantity": "1"}{"Product": "C", "Quantity": "1"}}  
  {"Header": 2, "Lines": {"Product": "E", "Quantity": "5"}}  
  {"Header": 3, "Lines": {"Product": "D", "Quantity": "4"}}  
  {"Header": 4, "Lines": {"Product": "A", "Quantity": "1"}{"Product": "C", "Quantity": "1"}}  
  {"Header": 5, "Lines": {"Product": "B", "Quantity": "3"}}  
  {"Header": 6, "Lines": {"Product": "D", "Quantity": "4"}}  
   
  The output should be (in whatever format you choose):  
  1: 1,0,1,0,0::1,0,1,0,0::0,0,0,0,0  
  2: 0,0,0,0,5::0,0,0,0,0::0,0,0,0,5  
  3: 0,0,0,4,0::0,0,0,0,0::0,0,0,4,0  
  4: 1,0,1,0,0::1,0,0,0,0::0,0,1,0,0  
  5: 0,3,0,0,0::0,3,0,0,0::0,0,0,0,0