/\*初始化\*/

begin model initialization function

set V\_s = 40 //可以修改s，S的值，观察成本的变化

set V\_S = 60

set T = 30 //查库周期，是固定的

set V\_Month = 120 //仿真长度，120个月

create 60 loads of load type L\_Product to P\_Start //初始库存水平

open "Cost.txt" for appending save result as V\_CSoutputfile //将输出结果保存到Cost.txt文件，appending是将结果写入并保留txt原本的内容

return true //每个初始化函数都要有一个返回值

end

/\*此进程中的load为初始化函数所产生的60个L\_Product\*/

begin P\_Start arriving //初始进程，将初始库存水平的货物放在仓库中存储，并在缓存区等待。

move into Q\_Warehouse

wait to be ordered on OL\_Wh

end

/\*查库 订购\*，此进程中的load为L\_Check\*/

begin P\_Check arriving //月初，检查库存水平，若低于最小库存，则需要进货

if OL\_Wh current loads < V\_s then

begin

inc j by 1 //对检查库存次数进行计数

set LA\_Buy = V\_S-OL\_Wh current loads //进货量

set V\_PrepareCost = V\_PrepareCost + 32 + (LA\_Buy\*3) /\*准备成本\*/

create LA\_Buy loads of load type L\_Product to P\_Supply //产生的load送到P\_Supply

send to P\_Wait

end

else send to die

end

/\*供货\*，此进程中的load为P\_Check所产生的L\_Product\*/

begin P\_Supply arriving //供应商供货，在到达销售商之前有一个到货提前期，在此之前，货物均在队列中存储，并在缓存区等待被调用

move into Q\_In

wait to be ordered on OL\_in

end

/\*进货到达之前的等待\*，此进程中的load为L\_Check\*/

begin P\_Wait arriving

set V\_Wait = uniform 22.5,7.5 //到货提前期的设置，服从22.5,7.5days的均匀分布

wait for V\_Wait days /\*到货提前期\*/

order all loads from OL\_in to P\_inWh //到货提前期结束，从OL\_in将所有的load调出到P\_inWh

end

/\*进货到达并入库，此进程中的load为P\_Check所产生的在OL\_in中等待的L\_Product\*/

begin P\_inWh arriving //货物进入到仓库存储，并在缓存区等待

move into Q\_Warehouse

set V\_Arrive = j\*30 + V\_Wait

order all loads from OL\_Shortage to die

wait to be ordered on OL\_Wh

end

/\*需求到达，此进程中的load为L\_CusOrder\*/

begin P\_Gendmd arriving

set V\_Order(i) = ac //ac 为仿真钟的现在的时刻，我们用V\_Order(i) 数组来记录每个顾客订单到达的时刻

set V\_Inventory(i) = OL\_Wh current loads //用V\_Inventory(i)记录订单i到达时的实时库存

set V\_LossQty(i) = OL\_Wh current backordered //用V\_LossQty(i)记录订单i到达时的实时缺货库存

set LA\_OrderQty = oneof(1:1,2:2,2:3,1:4) //本次需求量

set V\_OrderQty(i) = LA\_OrderQty //用V\_OrderQty(i)记录每张顾客订单的订单量，即需求量

move into Conv.sta3

travel to Conv.sta4

if V\_Inventory(i) >= V\_OrderQty(i) then /\*足够\*/

begin

set V\_Excess=OL\_Wh current loads-LA\_OrderQty //剩余量

if i = 1 then //i=1时，第一张订单到达，比较特殊，初始库存水平的存储时间就是0时刻与V\_Order(1)的间隔时间

begin

set V\_SumInventory = OL\_Wh current loads \* V\_Order(i)/2592000 + V\_SumInventory //总库存量的累加

inc i by 1 //i每次加1

end

else if i > 1 then //i大于1时，我们要讨论几种情况，因为由于月初的订货会导致库存的忽然增大，我们要讨论几种库存的情况

//有关讨论情况参考实验指导书

begin

if V\_Order(i) > V\_Arrive and V\_Order(i-1) >= V\_Arrive then

begin

set V\_SumInventory = ((V\_Order(i) - V\_Order(i-1))/2592000)\*V\_Inventory(i) + V\_SumInventory

end

else if V\_Order(i) > V\_Arrive and V\_Order(i-1) < V\_Arrive then

begin

if V\_Inventory(i-1) >= V\_OrderQty(i-1) then

begin

set V\_SumInventory = ((V\_Arrive/30) - (V\_Order(i-1)/2592000)) \* (V\_Inventory(i-1) - V\_OrderQty(i-1)) + V\_SumInventory

set V\_SumInventory = ((V\_Order(i)/2592000) - (V\_Arrive/30))\*V\_Inventory(i) + V\_SumInventory

end

else if V\_Inventory(i-1) < V\_OrderQty(i-1) and V\_Inventory(i-1) >= 0 then

begin

set V\_SumLoss = ((V\_Arrive/30) - (V\_Order(i-1)/2592000)) \* (V\_OrderQty(i-1)-V\_Inventory(i-1)) + V\_SumLoss

set V\_SumInventory = ((V\_Order(i)/2592000) - (V\_Arrive/30))\*V\_Inventory(i) + V\_SumInventory

end

else if V\_Inventory(i-1) < V\_OrderQty(i-1) and V\_LossQty(i-1) >= 0 then

begin

set V\_SumLoss = ((V\_Arrive/30) - (V\_Order(i-1)/2592000)) \* (V\_LossQty(i-1)+V\_OrderQty(i-1)) + V\_SumLoss

set V\_SumInventory = ((V\_Order(i)/2592000) - (V\_Arrive/30))\*V\_Inventory(i) + V\_SumInventory

end

end

else if V\_Order(i) <= V\_Arrive and V\_Order(i-1) < V\_Arrive then

begin

set V\_SumInventory = ((V\_Order(i) - V\_Order(i-1))/2592000)\*V\_Inventory(i) + V\_SumInventory

end

inc i by 1

end

order LA\_OrderQty loads from OL\_Wh to P\_Out

send to die

end

else if V\_Inventory(i) < V\_OrderQty(i) then /\*不够，有关成本计算的讨论参考实验指导书\*/

begin

if V\_Inventory(i) > 0 then

begin

set V\_SumInventory = ((V\_Order(i) - V\_Order(i-1))/2592000)\*V\_Inventory(i) + V\_SumInventory

end

else if V\_Inventory(i) = 0 and V\_Inventory(i-1) > 0 then

begin

set V\_SumLoss = ((V\_Order(i) - V\_Order(i-1))/2592000)\*V\_LossQty(i) + V\_SumLoss

//print "V\_Inventory:", i-1, V\_Inventory(i-1) to message

//print "V\_OrderQty:",i-1, V\_OrderQty(i-1) to message

end

else if V\_Inventory(i) = 0 and V\_Inventory(i-1) = 0 then

begin

set V\_SumLoss = ((V\_Order(i) - V\_Order(i-1))/2592000)\*(V\_LossQty(i-1)+V\_OrderQty(i-1)) + V\_SumLoss

end

set V\_Excess=0

inc i by 1

order LA\_OrderQty loads from OL\_Wh to P\_Out //缺货load在backorder等待被补齐

in case order not filled backorder on OL\_Wh

move into Q\_Shortage /\*进入该条件的均为短缺订单\*/

wait to be ordered on OL\_Shortage

end

end

/\*出库，此进程中的load为要出库的L\_Product\*/

begin P\_Out arriving

move into Conv.sta1

travel to Conv.sta2

send to die

end

/\*相关成本计算并输出到文件，此进程中的load为L\_Cost\*/

begin P\_Average arriving

set V\_AvgInventory = V\_SumInventory / V\_Month //计算每月平均库存

set V\_AvgLoss = V\_SumLoss / V\_Month //计算每月平均缺货量

set V\_AvgPrepareCost = V\_PrepareCost / V\_Month //各项平均成本，并输出

set V\_AvgHoldingCost = V\_AvgInventory \* 1

set V\_AvgShortageCost = V\_AvgLoss \* 5

set V\_AvgTotalCost = V\_AvgPrepareCost + V\_AvgShortageCost + V\_AvgHoldingCost

print "(", T "," ,V\_S "," V\_s ")" "\t" V\_AvgTotalCost "\t" "\t" V\_AvgPrepareCost "\t" "\t" V\_AvgHoldingCost "\t" "\t" V\_AvgShortageCost to V\_CSoutputfile //输出到文件

end