; h2 Core Code – each cj requests exactly one server, and the (minimally parallel) hps is degree 2

; In this special case, (one server per customer), the hps ulitization will agree with the model’s hps utilization

; ps processing uses interarrival mean 1 to simplify calculations

; The main purposes for the ps demos are 1) show effects as ps degree increases and 2) demo how

; to interpret ps server stats on a gpssW Report

; The gpssW Report display of the UTIL. value v for a STORAGE entity is NOT the utilization

; rho as we have defined it in this course; rather rho = (1 - v).

; Examples: for nps = 8 (8 homogeneous ps, v = .918 and sa$ps (for the STORAGE in the code below)

; = 7.34, reported as AVE.C. on STORAGE Report section.

; But, 1 - 7.34/8 = (1 - sa$ps/(number of ps) = (1 - 7.34/8) = 1 - .9175 = to 3 significant digits

; 1 - .918 = 0.082, that is, a little over 8% utilization.

; Similarly, for n ps = 2, v = .67 and sa$ps = 1.375, so rho = 1 - .67 = .33, so rho = 33% utilization.

; These calculations also hold for the special case of 1 server, for which rho = .64 (for 1000 tr,

; which has not yet converged to the theoretical limit of .67

;

clear

RN1\_seed EQU 934707

rmult RN1\_seed ; Seed gpss rand# generator RN1

ps\_ia\_mean EQU 1 ; ps Inter-arrival times mean

ps\_service\_mean EQU .67 ; Mean ps service duration

initial x$ps\_svrDuration,0 ; Store ps service duration values

ps\_ResidenceTime TABLE M1,0.5,0.5,20 ; Parallel server residence time distr

ps\_serviceDuration TABLE x$ps\_svrDuration,0.5,0.5,20 ; Parallel service duration distr

ps storage 2 ; Define parallel server count

init\_ps generate 0,,0,1,1 ; Initially create 2 ps barbers, just one time

enter ps,2 ; Allocate all ps barbers

terminate

cj generate (Exponential(1,0,ps\_ia\_mean)),,,100000 ; ss inter-arrivals distr.

queue ps\_res\_time ; Start cj residence duration stats

queue svr\_wait ; Start gathering service wait stats

gate SNE ps ; Wait here if all servers busy, otherwise, start a service

depart svr\_wait ; Finish gathering service wait stats

queue ps\_svr ; Start cj service duration stats

leave ps,1 ; Allocate exactly one ps server

; \*\*\* The following scenarios look at the effect of different service distributions

savevalue ps\_svrDuration,(Exponential(1,0,ps\_service\_mean)) ; Temp save service duration

tabulate ps\_serviceDuration ; Incr ps serviceDuration histo

advance x$ps\_svrDuration ; Do the service

enter ps,1 ; Finished service, and giver back server

depart ps\_svr ; Finish gathering service duration stats

depart ps\_res\_time ; Finish gathering cj residence duration stats

tabulate ps\_ResidenceTime ; Incr c’s residence duration

terminate 1 ; This cj leaves barbershop, & decr tc