Name ___________________________________________  Seat # ________ 1

1) Circle one of the following (-50 points):  I come to class at 8:00 am  12:45 pm
2) Please remove all hats. If it is part of your head, turn it around backwards.
3) Please note that if your work not legible, or if I cannot follow your logic at a glance, it will receive zero credit. If your code is scattered and messy I will not be able to follow it, so leave yourself time to recopy it if necessary. Better yet, just leave a lot of space between lines of code as you write it to fill in things later on that you forgot.
4) You MUST fully comment each and every line of code you write, specifying who was born, why they are waiting, why they are closing the gate, everything! Any line that is not commented will not be graded.
5) Exception: If you find large blocks of code that basically do the same thing, you can comment “same as above except for …” and not repeat the same comments again.

Problem 1) You are the traffic engineer for Disneyland, and have been tasked with determining how big a bus will be needed to adequately transport customers from the Parking Lot loading stations surrounding the Park, back to the main Park. You have decided to start by studying a 40 passenger bus.

The first bus leaves the Park at 6:00 a.m. and picks up patrons until 10:00 p.m. A second bus returns people to their cars, and is not involved in your study. The bus takes a mean of 5 minutes with a sigma of 1 minute to travel between Parking Lots, i.e. from the Park to Lot A, from Lot A to Lot B, etc. It takes the driver about three minutes to stop, load the passengers, and get back on the road (actually 3 minutes mean, exponentially distributed.) Unloading of passengers at the Park, their final destination, is exponential with a mean of 5 minutes. The driver then repeats the circuit.

If the bus fills up at any Lot during a run, the driver does not stop at the remaining Lots, but rather travels on past them and drives directly to the Park. Travel time between Lots is the same, the only time savings being due to not having to stop and pick up passengers.

Passengers arrive at the Bus Loading Stations (i.e. Lots) exponentially distributed with a mean of 2 passengers per minute.

Write a BOSS program to study this system.

Problem 2) You are the city engineer and have been tasked to study 3 local retention ponds in the area. Each of the ponds has the ability to hold 1,000,000 cubic ft of water before starting to empty over their spillways. The spillways have been carefully designed to release any excess above 1,000,000 cubic ft at the rate of 50*sqrt(excess) per minute.

Storms in the area are predicted to cross the ponds and drop the volumes of water at the times and rates shown in the figure. The times shown are in minutes after the start of the storms, and the rainfall is in cubic feet per minute into the pond listed.

Write a BOSS simulation to determine the maximum volume in each pond and the maximum flow out of each pond for a period of 300 minutes. Start the ponds empty.
PROGRAM
"Problem 1 - disney busses"

DEFINITION
roomonbus = 40;
letonbus = 0;
seated = 0;
peopleata = 0;
peopleatb = 0;
peopleatc = 0;
peopleatd = 0;
LABELS = {stationa, stationb, stationc, stationd, park};

CONTROL
STOPTIME = 16*60;

LOGIC
ARRIVE{TIME = 0, LIMIT = 1}; "born a bus"
stationa: WAIT{TIME = 0 MAX NORMAL(5,1)}; "travel time to station a"
   IF (roomonbus <= 0) THEN GOTO stationb;
      "if have seats, stop at a, else don't stop and go to b"
     letonbus = roomonbus MIN peopleata; "calculate how many can let on bus at a"
     seated = seated +letonbus; "calculate how many people now on bus"
    peopleata = peopleata -letonbus; "calculate how many people remain at lot a"
   WAIT{TIME = EXPD(3)}; "wait time to load passengers at a"

stationb: WAIT{TIME = 0 MAX NORMAL(5,1)}; "travel time to station b"
   IF (roomonbus <= 0) THEN GOTO stationc;
      "if have seats, stop at b, else don't stop and go to c"
     letonbus = roomonbus MIN peopleatb; "calculate how many can let on bus at b"
    seated = seated +letonbus; "calculate how many people now on bus"
   peopleatb = peopleatb -letonbus; "calculate how many people remain at lot b"
   WAIT{TIME = EXPD(3)}; "wait time to load passengers at b"

stationc: WAIT{TIME = 0 MAX NORMAL(5,1)}; "travel time to station c"
   IF (roomonbus <= 0) THEN GOTO stationd;
      "if have seats, stop at c, else don't stop and go to d"
     letonbus = roomonbus MIN peopleatc; "calculate how many can let on bus at c"
    seated = seated +letonbus; "calculate how many people now on bus"
   peopleatc = peopleatc -letonbus; "calculate how many people remain at lot c"
   WAIT{TIME = EXPD(3)}; "wait time to load passengers at c"

stationd: WAIT{TIME = 0 MAX NORMAL(5,1)}; "travel time to station d"
   IF (roomonbus <= 0) THEN GOTO park;
      "if have seats, stop at d, else don't stop and go to park"
     letonbus = roomonbus MIN peopleatd; "calculate how many can let on bus at d"
    seated = seated +letonbus; "calculate how many people now on bus"
   peopleatd = peopleatd -letonbus; "calculate how many people remain at lot d"
   WAIT{TIME = EXPD(3)}; "wait time to load passengers at d"
park: WAIT{TIME = 0 MAX NORMAL(5,1)}; "travel time to park"
   WAIT{TIME = EXPD(3)}; "unload passengers at park"
   seated = 0; "let everyone off bus"
   GOTO stationa; "make another trip"

ARRIVE{TIME = EXPD(1/2)}; "born passengers every 1/2 minute at station a"
   peopleata = peopleata + 1;
   DEPART{}; "leave"

ARRIVE{TIME = EXPD(1/2)}; "born passengers every 1/2 minute at station b"
   peopleatb = peopleatb + 1;
   DEPART{}; "leave"
ARRIVE\{TIME = EXPD(1/2)\}; "born passengers every 1/2 minute at station c"
peopleatc = peopleatc + 1;
DEPART{}; "leave"

ARRIVE\{TIME = EXPD(1/2)\}; "born passengers every 1/2 minute at station d"
peopleatd = peopleatd + 1;
DEPART{}; "leave"

END.

PROGRAM "retention ponds"

DEFINITION

LABELS = \{nextminutea, nextminuteb, nextminutec\};
pondcapacity = 1000000;
waterina = 0; waterinb = 0; waterinc = 0;
excessa = 0; excessb = 0; excessc = 0;
runoffa = 0; runoffb = 0; runoffc = 0;
maxwaterina = 0; maxwaterinb = 0; maxwaterinc = 0;
maxrunoffa = 0; maxrunoffb = 0; maxrunoffc = 0;

CONTROL
STOPTIME = 300; "minutes"

LOGIC

ARRIVE\{TIME = 0, LIMIT = 1\}; "born storm over pond a"
nextminutea:
IF CLOCKTIME <= 50 THEN waterina = waterina + 10000; "put a minute's rain into a if before 50 mins"
IF CLOCKTIME IN\[50,140\] THEN waterina = waterina + 60000; "put minute of rain into a if between 50 and 140"
WAIT\{TIME = 1\}; "wait a minute"
maxwaterina = waterina MAX maxwaterina;
IF waterina >= pondcapacity THEN excessa = waterina - pondcapacity ELSE excessa = 0;
"calculate excess in pond a"
runoffa = 50*excessa^0.5; "calculate runoff from pond a"
maxrunoffa = maxrunoffa MAX runoffa; "keep track of max runoff from pond a"
waterina = waterina - runoffa; "calculate reduction of water in a after runoff out of a"
waterinb = waterinb + runoffa; "calculate added water in b due to runoff from a"
GOTO nextminutea; "go back and see what happens during next minute"

ARRIVE\{TIME = 0, LIMIT = 1\}; "born storm over pond b"
nextminuteb:
IF CLOCKTIME IN\[50,90\] THEN waterinb = waterinb + 20000; "put minute of rain into b if between 50 and 90"
IF CLOCKTIME IN\[90,160\] THEN waterinb = waterinb + 80000; "put minute of rain into b if between 90 and 160"
WAIT\{TIME = 1\}; "wait a minute"
maxwaterinb = waterinb MAX maxwaterinb;
IF waterinb >= pondcapacity THEN excessb = waterinb - pondcapacity ELSE excessb = 0;
"calculate excess in pond b"
runoffb = 50*excessb^0.5; "calculate runoff from pond b"
maxrunoffb = maxrunoffb MAX runoffb; "keep track of max runoff from pond b"
waterinb = waterinb - runoffb; "calculate reduction of water in b after runoff out of b"
waterinc = waterinc + runoffb; "calculate added water in c due to runoff from b"
GOTO nextminuteb; "go back and see what happens during next minute"

ARRIVE\{TIME = 0, LIMIT = 1\}; "born storm over pond c"
nextminutec:
IF CLOCKTIME IN\[80,150\] THEN waterinc = waterinc + 70000; "put minute
of rain into c if between 80 and 150"

IF CLOCKTIME IN[150,180] THEN waterinc = waterinc + 100000; "put minute
of rain into c if between 150 and 180"

WAIT{TIME = 1}; "wait a minute"

maxwaterinc = waterinc MAX maxwaterinc;

IF waterinc >= pondcapacity THEN excessc = waterinc-pondcapacity ELSE excessc = 0;

"calculate excess in pond c"

runoffc = 50*excessc^0.5; "calculate runoff from pond c"

maxrunoffc = maxrunoffc MAX runoffc; "keep track of max runoff from pond c"

waterinc = waterinc - runoffc; "calculate reduction of

water in c after runoff out of c"

GOTO nextminutec; "go back and see what happens during next minute"

END.