1) DO NOT put your completed work on the desk next to you or anywhere else where it can be seen. If I come by and see it I will confiscate it and give you zero credit for that problem. Or worse, someone will copy your work and I won’t be able to tell which of you did the copying and accuse the wrong person. Place it face down on your desk under your existing work. Please take this instruction very seriously.

2) Please remove your hat. If it is part of your head, turn it around backwards.

3) Please note that if your work is not legible, if it uses a font size smaller than 12 point, or if I cannot follow your logic at a glance, it will receive zero credit. Annotate all lines of code.

Problem 1) (30 points) An underpass under the Union Pacific rails has been proposed for the intersection of Villa Maria and Finfeather Road in Bryan Texas. The purpose of this underpass is to allow emergency vehicles zero-delay access to the South side of town. Before spending a million dollars on this underpass, I have been asked to determine what delays are actually being caused by train traffic, and their impact on this traffic. All statistical information listed below is normally distributed, unless otherwise noted. Trains come across Villa Maria at exactly every two hours, all day long, starting at midnight. Most of the trains travel through the intersection at between 10 to 15 mph. However, 10% of the trains must stop to switch to another track, thereby blocking the intersection for a total of from five to ten minutes. Trains consist of from 20 to 70 cars, with a mean car length of from 50 to 86 feet. Gates at the intersection close 30 seconds before the arrival of the train, and open 20 seconds after it clears the intersection. Emergency vehicles are called through the intersection 8 times a day exponentially distributed. Write a BOSS program to determine how many times in a one year period an emergency vehicle will be delayed by a train, and the average delay time.

Problem 2) (30 points) Our company manufactures aircraft engine bearing pads from steel and aluminum plates bonded together. One bearing pad consists of three steel plates, separated by two aluminum plates to form the pad. The basic cost of a steel plate is $10 and the aluminum plates cost $12 each. The plates are bonded together with a resin/aluminum sintering powder. Once the plates are glued and stacked (steel, al, steel, al, steel) they are compressed and heated to 1400 degrees F to form the final pad. The steel plate thickness ranges from 0.490” to 0.530”, the aluminum plates range from 0.460” to 0.510” and the final thickness of the bonding resin (4 layers) is 0.005” to 0.010” per layer.

Pads over 2.56” in final thickness must be discarded. Pads between 2.51” and 2.56” can be ground down. Grinding removes 0.01” of thickness per pass and costs $4.00 per pass. Pads between 2.48” and 2.51” thick are acceptable and need no further work. Pads between 2.42” and 2.48” can be shimmed up with 0.01” shims which cost $1.00/shim. Pads thinner than 2.42” must be discarded. Determine the total cost to make 1000 shims, the total grinding and shimming costs, and the total number of pads discarded.

Problem 3) (15 points) For the program listed below, determine the final value of COUNT. Incidentally, the program DOES run. I ran it.

```
PROGRAM
"422 exam B1"
DEFINITION
count = 10;
LABELS = {dog};
maingate: GATE = {STATUS=CLOSE};
CONTROL
STOPTIME=10;
RANDOMIZE = ON;
LOGIC "logic for all arriving traffic"
count = count + 2;
ARRIVE {TIME=0, LIMIT = 5};
SETGATE{NAME = maingate, STATUS = CLOSE};
WAIT (TIME = 5);
SETGATE{NAME = maingate, STATUS = OPEN};
count = count + 2;
WAIT (TIME=30);
DEPART { }
;
count = count + 2;
ARRIVE {TIME = 1, LIMIT = 5};
TESTGATE{NAME = maingate, IFCLOSED = dog};
WAIT (TIME = 5);
dog: SETGATE{NAME = maingate, STATUS = OPEN};
count = count + 2;
DEPART { }
;
END.
```

Problem 4) (25 points) ABOUT how much interest will I save if I pay an extra $100/month on my home loan mortgage? My loan is for $80,000, 15 years, monthly payments, 6% annual interest rate. It doesn’t have to be exact, but should be pretty close.
<table>
<thead>
<tr>
<th>n</th>
<th>P/A</th>
<th>P/F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>1.000000</td>
<td>0.995025</td>
</tr>
<tr>
<td>1</td>
<td>0.995025</td>
<td>1.005000</td>
</tr>
<tr>
<td>2</td>
<td>1.980509</td>
<td>0.503753</td>
</tr>
<tr>
<td>3</td>
<td>2.970248</td>
<td>0.336672</td>
</tr>
<tr>
<td>4</td>
<td>3.950496</td>
<td>0.253133</td>
</tr>
<tr>
<td>5</td>
<td>4.925866</td>
<td>0.203301</td>
</tr>
<tr>
<td>6</td>
<td>5.896384</td>
<td>0.169595</td>
</tr>
<tr>
<td>7</td>
<td>6.862074</td>
<td>0.145729</td>
</tr>
<tr>
<td>8</td>
<td>7.822955</td>
<td>0.127829</td>
</tr>
<tr>
<td>9</td>
<td>8.779604</td>
<td>0.113907</td>
</tr>
<tr>
<td>10</td>
<td>9.730412</td>
<td>0.102727</td>
</tr>
<tr>
<td>11</td>
<td>10.677077</td>
<td>0.093659</td>
</tr>
<tr>
<td>12</td>
<td>11.618932</td>
<td>0.086066</td>
</tr>
<tr>
<td>13</td>
<td>12.551551</td>
<td>0.079642</td>
</tr>
<tr>
<td>14</td>
<td>13.487080</td>
<td>0.074136</td>
</tr>
<tr>
<td>15</td>
<td>14.416825</td>
<td>0.069364</td>
</tr>
<tr>
<td>16</td>
<td>15.339925</td>
<td>0.065189</td>
</tr>
<tr>
<td>17</td>
<td>16.268332</td>
<td>0.061050</td>
</tr>
<tr>
<td>18</td>
<td>17.172768</td>
<td>0.057823</td>
</tr>
<tr>
<td>19</td>
<td>18.082356</td>
<td>0.055300</td>
</tr>
<tr>
<td>20</td>
<td>18.987419</td>
<td>0.052686</td>
</tr>
<tr>
<td>21</td>
<td>19.887979</td>
<td>0.050282</td>
</tr>
<tr>
<td>22</td>
<td>20.784059</td>
<td>0.048114</td>
</tr>
<tr>
<td>23</td>
<td>21.675681</td>
<td>0.046135</td>
</tr>
<tr>
<td>24</td>
<td>22.565483</td>
<td>0.044321</td>
</tr>
<tr>
<td>25</td>
<td>23.445130</td>
<td>0.042652</td>
</tr>
<tr>
<td>26</td>
<td>24.324018</td>
<td>0.041112</td>
</tr>
<tr>
<td>27</td>
<td>25.198028</td>
<td>0.039686</td>
</tr>
<tr>
<td>28</td>
<td>26.067689</td>
<td>0.038362</td>
</tr>
<tr>
<td>29</td>
<td>27.930454</td>
<td>0.036989</td>
</tr>
<tr>
<td>30</td>
<td>28.513238</td>
<td>0.035979</td>
</tr>
<tr>
<td>31</td>
<td>29.650800</td>
<td>0.034931</td>
</tr>
<tr>
<td>32</td>
<td>30.503264</td>
<td>0.033895</td>
</tr>
<tr>
<td>33</td>
<td>30.935162</td>
<td>0.032947</td>
</tr>
<tr>
<td>34</td>
<td>31.915548</td>
<td>0.032056</td>
</tr>
<tr>
<td>35</td>
<td>32.305371</td>
<td>0.031215</td>
</tr>
<tr>
<td>36</td>
<td>32.877106</td>
<td>0.030422</td>
</tr>
<tr>
<td>37</td>
<td>33.702504</td>
<td>0.029671</td>
</tr>
<tr>
<td>38</td>
<td>34.529854</td>
<td>0.028906</td>
</tr>
<tr>
<td>39</td>
<td>35.353089</td>
<td>0.028286</td>
</tr>
<tr>
<td>40</td>
<td>36.172288</td>
<td>0.027646</td>
</tr>
<tr>
<td>41</td>
<td>36.978291</td>
<td>0.027036</td>
</tr>
<tr>
<td>42</td>
<td>37.798300</td>
<td>0.026456</td>
</tr>
<tr>
<td>43</td>
<td>38.605274</td>
<td>0.025853</td>
</tr>
<tr>
<td>44</td>
<td>39.408232</td>
<td>0.025375</td>
</tr>
<tr>
<td>45</td>
<td>40.207196</td>
<td>0.024871</td>
</tr>
<tr>
<td>46</td>
<td>41.002185</td>
<td>0.024389</td>
</tr>
<tr>
<td>47</td>
<td>41.793219</td>
<td>0.023927</td>
</tr>
<tr>
<td>48</td>
<td>42.580318</td>
<td>0.023486</td>
</tr>
<tr>
<td>49</td>
<td>43.363500</td>
<td>0.023061</td>
</tr>
<tr>
<td>50</td>
<td>44.142786</td>
<td>0.022654</td>
</tr>
</tbody>
</table>

**Interest rate per compounding period = 0.005**
1) DO NOT put your completed work on the desk next to you or anywhere else where it can be seen. If I come by and see it I will confiscate it and give you zero credit for that problem. Or worse, someone will copy your work and I won’t be able to tell which of you did the copying and accuse the wrong person. Place it face down on your desk under your existing work. Please take this instruction very seriously.

2) Please remove your hat. If it is part of your head, turn it around backwards.

3) Please note that if your work is not legible, if it uses a font size smaller than 12 point, or if I cannot follow your logic at a glance, it will receive zero credit. Annotate all lines of code.

Problem 1) (30 points) Our company manufactures aircraft engine bearing pads from steel and aluminum plates bonded together. One bearing pad consists of three steel plates, separated by two aluminum plates to form the pad. The basic cost of a steel plate is $20 and the aluminum plates cost $22 each. The plates are bonded together with a resin/aluminum sintering powder. Once the plates are glued and stacked (steel, al, steel, al, steel) they are compressed and heated to 800 degrees F to form the final pad. The steel plate thickness ranges from 0.460” to 0.510”, the aluminum plates range from 0.490” to 0.530” and the final thickness of the bonding resin (4 layers) is 0.005” to 0.010” per layer.

Pads over 2.56” in final thickness must be discarded. Pads between 2.51” and 2.56” can be ground down. Grinding removes 0.01” of thickness per pass and costs $4.00 per pass. Pads between 2.48” and 2.51” thick are acceptable and need no further work. Pads between 2.42” and 2.48” can be shimmed up with 0.01” shims which cost $1.00/shim. Pads thinner than 2.42” must be discarded. Determine the total cost to make 1000 shims, the total grinding and shimming costs, and the total number of pads discarded.

Problem 2) (30 points) An underpass under the Santa Fe rails has been proposed for the intersection of Villa Maria and Finsheather Road in Bryan Texas. The purpose of this underpass is to allow emergency vehicles zero-delay access to the South side of town. Before spending a million dollars on this underpass, I have been asked to determine what delays are actually being caused by train traffic, and their impact on this traffic. All statistical information listed below is normally distributed, unless otherwise noted. Trains come across Villa Maria at exactly every hour, all day long, starting at midnight. Most of the trains travel through the intersection at between 12 to 18 mph. However, 5% of the trains must stop to switch to another track, thereby blocking the intersection for a total of from eight to twelve minutes. Trains consist of from 30 to 50 cars, with an average car length of from 60 to 86 feet. Gates at the intersection close 30 seconds before the arrival of the train, and open 20 seconds after it clears the intersection. Emergency vehicles are called through the intersection 8 times a day exponentially distributed. Write a BOSS program to determine how many times in a one year period emergency vehicles will be delayed by trains, and the total delay time.

Problem 3) (25 points) ABOUT how much interest will I save if I pay an extra $100/month on my home loan mortgage? My loan is for $180,000, 15 years, monthly payments, 12% annual interest rate. It doesn’t have to be exact, but should be pretty close.

Problem 4) (15 points) For the program listed below, determine the final value of COUNT. Incidentally, the program DOES run. I ran it.

PROGRAM
"422 exam B1"
DEFINITION
count = 20;
LABELS = {dog};
maingate: GATE = {STATUS=CLOSED};
CONTROL
STOPTIME=10;
RANDOMIZE = ON;
LOGIC "logic for all arriving traffic"
count = count + 3;
ARRIVE {TIME=0, LIMIT = 5};
SETGATE{NAME = maingate, STATUS = CLOSE};
WAIT {TIME = 5};
SETGATE{NAME = maingate, STATUS = OPEN};
count = count + 3;
WAIT {TIME=30};
DEPART { };
count = count + 3;
ARRIVE {TIME = 1, LIMIT = 5};
TESTGATE{NAME = maingate, IFCLOSED = dog};
WAIT {TIME = 5};
dog: SETGATE{NAME = maingate, STATUS = OPEN};
count = count + 3;
DEPART { }; END.
PROGRAM
"Quiz B - Trains and ambulances"
"NOTE - different exams had different numbers - these may not be exactly as yours"

DEFINITION
  numbercars = 0;
  speed = 0;
  avgcarlength = 0;
  trainlength = 0;
  timetoclear = 0;
  timeintointersection = 0;
  timeoutofintersection = 0;
  mydelay = 0;
  totaldelays = 0;
  totaltimedelay = 0;
  averagedelay = 0;
  LABELS = {longwait, shortwait, delayed};

  crossarm: GATE={STATUS=OPEN};

CONTROL
  STOPTIME = 365*24*60*60; "seconds"
  RANDOMIZE = ON;

LOGIC

  "Logic for trains"
  ARRIVE {TIME = 2*60*60}; "train arrives and closes gate"
    numbercars = 0 MAX NORMAL(45, (70-20)/6); "determine number of cars in this train. non-integer OK"
    avgcarlength = 0 MAX NORMAL(40, (50-30)/6); "determine average length of a car in this train (feet)"
    trainlength = numbercars*avgcarlength; "determine total length of this train (feet)"
    speed = 0 MAX NORMAL(12.5, (15-10)/6)* 5280/3600; "determine speed of this train (feet/sec)"
    timetoclear = trainlength/speed; "determine time for this train to clear intersection"
  SETGATE{NAME = crossarm, STATUS = CLOSE}; "close crossarm"
  IF RANDOM <= 0.1 THEN GOTO longwait ELSE GOTO shortwait;

longwait:
  WAIT{TIME = 30}; "gate closes 30 seconds before arrival of train"
  WAIT{TIME = 0 MAX NORMAL((5+10)/2, (10-5)/6)}; "wait for train to change tracks"
  WAIT{TIME = 20}; "gate opens 20 seconds after train clears"
  SETGATE{NAME = crossarm, STATUS = OPEN}; "open crossarm"
  DEPART; "train departs"

shortwait:
  WAIT{TIME = 30}; "gate closes 30 seconds before arrival of train"
  WAIT{TIME = timetoclear}; "gate remains closed while train passes"
  WAIT{TIME = 20}; "gate opens 20 seconds after train clears"
  SETGATE{NAME = crossarm, STATUS = OPEN}; "open crossarm"
  DEPART; "train departs"

  "Logic for traffic"
  ARRIVE {TIME=EXPDT(24*60*60/8)}; "ambulance arrives"
  timeintointersection = CLOCKTIME; "set time into intersection"
  TESTGATE{NAME = crossarm}; "stop if crossarm is down"
timeoutofintersection = CLOCKTIME; "set time out of intersection"
mydelay = timeoutofintersection - timeintointersection; "calculate delay due
to train"
IF mydelay >= 0.001 THEN GOTO delayed ELSE DEPART{}; "if delay is nil, depart"
delayed:
totaldelays = totaldelays + 1; "increment total delays"
totaltimedelay = totaltimedelay + mydelay; "increment delay times"
averagedelay = totaltimedelay/totaldelays; "calculate new average delay time"
DEPART;

END.
PROGRAM
"Quiz B1 - Manufacture of bearing pads"
"NOTE: different people had different exams. These numbers may not be exactly
the same as yours"
"Things you should know from Stat 211:
 If you pick 3 steel plates at random, they will not all have the same
thickness.
 These thicknesses will be normally distributed, not uniformly or continuously
distributed."
"Practical uses of this program:
 1) Get costs for 1000 pads using the current plate tolerances.
 2) Pay higher costs for new plates with tighter tolerances and see if you
can save money by not having to throw away as many pads.
 3) Look at the output and see if you have 2500 grinds, and 2 shims.
This would mean you could save money by ordering plates with the same
tolerance costs, but having a thinner mean thickness, and shift the
expensive grinding operation towards the less expensive shimming operation.
After awhile, you will find that you run out of abililty to shim, and the
cost again goes up.
 4) You could do this by reducing the mean plate thicknesses by 0.01, and
rerunning
the problem, reducing them another 0.01 and rerunning, etc."

DEFINITION
  goodpads = 0;
  badpads = 0;
  steel1 = 0; steelt2 = 0; steelt3 = 0;
  alumnt1 = 0; alumnt2 = 0;
  resin1 = 0; resin2 = 0; resin3 = 0; resin4 = 0;
  thickness = 0;
  cost = 0;

LABELS = {again, Tcheck, grind, goodpad, shim, badpad};

CONTROL
  STOPTIME = 1; "seconds"
  RANDOMIZE = ON;

LOGIC
  ARRIVE {TIME = 0, LIMIT = 1}; "arrive person to assemble pads"
again: IF goodpads >= 1000 THEN WAIT(TIME = 2); "After 1000 good plates
assembled, wait so system wont go empty"

  steelt1 = 0 MAX NORMAL((0.53+0.49)/2, (0.53-0.49)/6); "determine thickness of
  steelt2 = 0 MAX NORMAL((0.53+0.49)/2, (0.53-0.49)/6); "determine thickness of
  steelt3 = 0 MAX NORMAL((0.53+0.49)/2, (0.53-0.49)/6); "determine thickness of
  alumnt1 = 0 MAX NORMAL((0.51+0.46)/2, (0.51-0.46)/6); "determine thickness of
  alumnt2 = 0 MAX NORMAL((0.51+0.46)/2, (0.51-0.46)/6); "determine thickness of
  resin1 = 0 MAX NORMAL((0.005+0.010)/2, (0.010-0.005)/6); "determine
  thickness of resin1"
  resin2 = 0 MAX NORMAL((0.005+0.010)/2, (0.010-0.005)/6); "determine
  thickness of resin2"
  resin3 = 0 MAX NORMAL((0.005+0.010)/2, (0.010-0.005)/6); "determine
thickness of resin3
    resin4 = 0 MAX NORMAL((0.005+0.010)/2, (0.010-0.005)/6); "determine
thickness of resin4"
    cost = cost + 54.00; "add basic cost for this pad, whether good or bad"

    thickness =
    steelt1+steelt2+steelt3+alumnt1+alumnt2+resin1+resin2+resin3+resin4;
    "calculate thickness of pad"

Tcheck: IF thickness >= 2.56 THEN GOTO badpad; "pad too thick to work with"
    IF thickness IN[2.51, 2.56] THEN GOTO grind; "pad needs grinding"
    IF thickness IN[2.48, 2.51] THEN GOTO goodpad; "pad OK as is"
    IF thickness IN[2.42, 2.48] THEN GOTO shim; "pad needs shimming"
    IF thickness <= 2.42 THEN GOTO badpad; "pad too thin to work with"

badpad:   badpads = badpads+1; "add 1 to junk pile"
    GOTO again; "make another pad"

grind:    thickness = thickness - 0.01; "grind 0.01 off of pad thickness"
    cost = cost + 4.00; "add one grinding operation cost"
    GOTO Tcheck; "go back and remeasure pad thickness"

goodpad:  goodpads = goodpads + 1; "add 1 to goodpads"
    GOTO again; "make another pad"

shim:     thickness = thickness + 0.01; "add 0.01 to pad thickness"
    cost = cost + 1.00; "add cost of one shim"
    GOTO Tcheck; "go back and remeasure pad thickness"

END.
A = ?

\[ n = 15 \times 12 = 180 \text{ months} \]

\[ i = \frac{6\%}{12} = 0.5\% \]

\[ 80000 = A \left[ \frac{P}{A}, 0.5\%, 180 \right] \]

\[ \frac{118.504}{\text{118,504}} \]

\[ \Rightarrow A = \$675.09 \text{ / month} \]

Now \( A = 675.09 + 100 = \$775.09 \text{ / month} \).

\[ 80000 = 775.09 \left[ \frac{P}{A}, 0.5\%, n \right] \]

\[ \Leftrightarrow 103.214 = \frac{(1.005)^n - 1}{0.005 \cdot (1.005)^n} \]

\[ \Rightarrow n = 146 \text{ months} \]

Then, \( \text{Money saved} = 180 \times 675.09 - 146 \times 775.09 \) \[ = \$8353 \]


\[ 180,000 \]

\[ n = 15 \times 12 = 180 \text{ months} \]

\[ i = \frac{12\%}{12} = 1\% \]

\[ 180,000 = A \left[ \frac{P}{A}, 1\%, 180 \right] \]

\[ A = \$2160.3 \]

New \( A \) = 2160.3 + 100 = \$2260.3

\[ 180,000 = 2260.3 \left[ \frac{P}{A}, 1\%, n \right] \]

\[ 79.64 = \frac{(0.01)^n - 1}{(0.01)(1.01)^n} \]

\[ n = 160 \text{ months} \]

Then, Money Saved = 180 \times 2160.3 - 160 \times 2260.3

\[ = \$27,206 \]
MOR/DS 1.00

Date: 11/16/05
Time: 10:53:03

c:\junk\prob3.bos

MODEL DESCRIPTION

******************************************************************************

PROGRAM
"422 exam B1"

DEFINITION

    count = 10;
    LABELS = {dog};
    maingate: GATE = {STATUS=CLOSED};

CONTROL

    STOPTIME=10;
    RANDOMIZE = ON;

LOGIC "logic for all arriving traffic"

    [ 1 ]
      count = count + 2;
    [ 2 ]
    [ 3 ]
      ARRIVE {TIME=0, LIMIT = 5};
    [ 4 ]
      SETGATE {NAME = maingate, STATUS = CLOSE};
    [ 5 ]
      WAIT {TIME = 5};
    [ 6 ]
      SETGATE {NAME = maingate, STATUS = OPEN};
    [ 7 ]
      count = count + 2;
    [ 8 ]
      WAIT {TIME=30};
    [ 9 ]
      DEPART { };  
    [10 ]
      count = count + 2;
    [11 ]
      ARRIVE {TIME = 1, LIMIT = 5};
    [12 ]
      TESTGATE {NAME = maingate, IFCLOSED = dog};
    [13 ]
      WAIT {TIME = 5};
    [14 ]
      dog: SETGATE {NAME = maingate, STATUS = OPEN};
    [15 ]
      count = count + 2;
    [16 ]
      DEPART { };
    [17 ]
    [18 ]
END.

-------------------------------------------------------------------------------

Simulation Clock : 10.00

Block Information

<table>
<thead>
<tr>
<th>Stmt</th>
<th>Line</th>
<th>Label</th>
<th>Name</th>
<th>Total</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td>ASSIGNMENT</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Page 1 of 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 16

ARRIVE 5 0
SETGATE 5 0
WAIT 5 0
SETGATE 5 0
ASSIGNMENT 5 0
WAIT 5 5
DEPART 0 0
ASSIGNMENT 0 0
ARRIVE 5 0
TESTGATE 5 0
WAIT 4 0
SETGATE 5 0
ASSIGNMENT 5 0
DEPART 5 5

------------------- Gates -------------------

MAINGATE
Default Queue Statistics
No entries.

------------------- Global Variables -------------------
COUNT = 30.00

Compile time: 0.05 (Secs.)
Run time : 0.00 (Secs.)
Total memory available : 217082
Memory used by model : 2459
Maximum dynamic memory used: 1320
Total memory used : 3779
PROGRAM
"422 exam B1"

DEFINITION
count = 20;
LABELS = {dog};
maingate: GATE = {STATUS=CLOSED};

CONTROL
STOPTIME=10;
RANDOMIZE = ON;

LOGIC "logic for all arriving traffic"
[ 1]
[ 2]    count = count + 3;
[ 3]    ARRIVE {TIME=0, LIMIT = 5};
[ 4]    SETGATE{NAME = maingate, STATUS = CLOSE};
[ 5]    WAIT{TIME = 5};
[ 7]    count = count + 3;
[ 8]    WAIT{TIME=30};
[ 9]    DEPART{ }; 
[10]    count = count + 3;
[11]    ARRIVE{TIME = 1, LIMIT = 5};
[12]    TESTGATE{NAME = maingate, IFCLOSED = dog};
[13]    WAIT{TIME = 5};
[14]    dog: SETGATE{NAME = maingate, STATUS = OPEN};
[15]    count = count + 3;
[16]    DEPART { }; 
[17]
[18]    END.

------------------------------------------
Simulation Clock : 10.00

Block Information

<table>
<thead>
<tr>
<th>Stmt</th>
<th>Line</th>
<th>Label</th>
<th>Name</th>
<th>Total</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td>ASSIGNMENT</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Page 1 of 2
| 2 | 3 | ARRIVE  | 5 | 0 |
| 3 | 4 | SETGATE | 5 | 0 |
| 4 | 5 | WAIT    | 5 | 0 |
| 5 | 6 | SETGATE | 5 | 0 |
| 6 | 7 | ASSIGNMENT | 5 | 0 |
| 7 | 8 | WAIT    | 5 | 0 |
| 8 | 9 | DEPART  | 0 | 0 |
| 9 | 10| ASSIGNMENT | 0 | 0 |
| 10| 11| ARRIVE  | 5 | 0 |
| 11| 12| TESTGATE| 5 | 0 |
| 12| 13| WAIT    | 4 | 0 |
| 13| 14| DOG     |    |   |
| 14| 15| SETGATE | 5 | 0 |
| 15| 16| ASSIGNMENT | 5 | 0 |
|    | 16| DEPART  | 5 | 5 |

------------------------ Gates ------------------------

MAINGATE
Default Queue Statistics
No entries.

------------------------ Global Variables ------------------------
COUNT = 50.00

Compile time: 0.00 (Secs.)
Run time : 0.06 (Secs.)
Total memory available : 217082
Memory used by model : 2459
Maximum dynamic memory used: 1320
Total memory used : 3779