READ THE FOLLOWING GENERAL EXAMINATION RULES:

1) Do not put your completed work anywhere that it can be seen. If any part of your work can be seen by others it will be confiscated and you will not be permitted to rework those problems. Place any pages of your work face down on your desk under your existing work, not on the floor next to you where it is visible.

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

3) If your work is not legible, or if I cannot follow your logic at a glance, it will receive no credit. This paper must be written to acceptable engineering standards for credit. Please take this seriously as it will affect your grade.

4) You may work on the front or back of this paper. Just note if work is on the back.

5) You can use your own paper or paper supplied at the front of the room.

6) You MUST specify what you are doing every step of the way. ONLY if I can follow what you are trying to do and where you are getting your numbers from, will you receive partial credit should you go off track.

7) Write big and use lots of paper, leaving me room to grade your paper. If there is no room to tell you why points were deducted, I will only show you the point deduction and let you try and figure out why.

I have read and understand all of the above instructions: ______________ (Initials)

Ethical Standards:

Upon accepting admission to Texas A&M University, a student immediately assumes a commitment to uphold the Honor Code, to accept responsibility for learning, and to follow the philosophy and rules of the Honor System. Students will be required to state their commitment on examinations, research papers, and other academic work. Ignorance of the rules does not exclude any member of the TAMU community from the requirements or the processes of the Honor System.

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this exam."

__________________________________________
Signature of student

Please do not open this exam until you are told to do so.
Problem 1) (25 points) My bank is a really nice bank. Today is November 30th, 2013. Normally if I miss a payment, they charge me a really hefty late penalty. However, they just emailed me and said that since it was Christmas, and since they were really nice guys, I could skip my upcoming end of December payment without a late charge. Having maxed out the card at $40,000, I had already decided to cut it up and stop charging things, since they were charging me 18% interest. I had intended pay off my balance in the next 24 months in equal installments, but due to their offer I think I might go ahead skip the December 31st payment, and then pay them off in the following 24 months. How much extra interest will I end up paying if I decide to skip my December payment?
Problem 1) (25 points) My bank is a really nice bank. Today is November 30th, 2013. Normally if I miss a payment, they charge me a really hefty late penalty. However, they just emailed me and said that since it was Christmas, and since they were really nice guys, I could skip my upcoming end of December payment without a late charge. Having maxed out the card at $40,000, I had already decided to cut it up and stop charging things, since they were charging me 18% interest. I had intended pay off my balance in the next 24 months in equal installments, but due to their offer I think I might go ahead skip the December 31st payment, and then pay them off in the following 24 months. How much extra interest will I end up paying if I decide to skip my December payment?

\[ i = \frac{18.9\%}{12} = 1.5\% \]

\[ 0.015(40,000) = \$600/\text{month} \]

\[
\begin{align*}
40,000\left(\frac{A}{P}, 1.5\%, 24\right) &= 40,000(0.0499) = \$1996 \\
1996(24) &= \$47,904
\end{align*}
\]

\[
40,000\left(\frac{A}{P}, 1.5\%, 25\right)\left(\frac{A}{F}, 1.5\%, 24\right)
\]

\[
= 40,000(1.4509)(0.0349) \\
= 2025.4564 \\
2025.4564(24) = \$48,610.9536
\]

\[ \$48,610.9536 - \$47,904 = \$706.95 \]
Problem 1) (25 points) My bank is a really nice bank. Today is November 30th, 2013. Normally if I miss a payment, they charge me a really hefty late penalty. However, they just emailed me and said that since it was Christmas, and since they were really nice guys, I could skip my upcoming end of December payment without a late charge. Having maxed out the card at $40,000, I had already decided to cut it up and stop charging things, since they were charging me 18% interest. I had intended pay off my balance in the next 24 months in equal installments, but due to their offer I think I might go ahead skip the December 31st payment, and then pay them off in the following 24 months. How much extra interest will I end up paying if I decide to skip my December payment?

\[
-40k + \frac{0.0304}{1.015^{24}} + 24 = 0
\]
\[
A_1 = 1.997k
\]

\[
-40k + \frac{0.0304}{1.015^{25}} + 25 = 0
\]
\[
A_2 = 2.027k
\]

\[
A_1(24) = \$47.927k \ \text{total}
\]
\[
A_2(24) = \$48.647k \ \text{total}
\]
\[
\$719.98 \ \text{extra interest if skip Dec payment}
\]
Problem 2) (25 points) Last night I had the strangest dream. I dreamed that sheep entered through a door in a barn according to a uniform distribution with a minimum time of 9 and a maximum of 12 seconds. 25% of the sheep were black, the rest were white. After entering, the sheep ran over to a fence. It took them from 8 to 13 seconds to get to it, distributed normally. They then proceeded to jump the fence which took 6 seconds. Once over the fence, 50% of the black sheep and 20% of the white sheep were turned into mutton. The rest ran out the back door and escaped. Each sheep processed yielded between 50 and 80 pounds of mutton, distributed normally.

Pigs also entered the barn, exponentially distributed with a mean of 35 seconds. They were all pink. They also ran up to the fence, but ran around it, taking between 40 and 50 seconds distributed normally. Once past the fence, 10% of the pigs were caught and turned into bacon, and the rest ran towards the exit. However, not being very smart, 30% of those returned to the gate and went through the system in the same manner as before. The rest went escaped. Each pig yielded a mean of 50 pounds of bacon with a sigma of 10 pounds.

Write a BOSS program to determine how much bacon and how much mutton I had after 8 hours of dreaming.
PROGRAM
"sheep and pigs"

DEFINITION

ATTRIBUTES = {color = 0}; "give each sheep their own color"
fence :RESOURCE = {}; 
mutton = 0; 
bacon = 0; 

LABELS = {blacksheep, dummy, processpigs, slaughter};

CONTROL
STOPTIME=8*60*60; "run 8 hours in seconds"
RANDOMIZE = ON;

LOGIC

ARRIVE {TIME=CUNIFORM(9,12)}; "born black and white sheep"
IF RANDOM <= 0.25 THEN color = 0 ELSE color = 1; "set 0 = black"

"Note that you MUST use an ATTRIBUTE if you set their colors here, because otherwise the next sheep coming shortly thereafter will change everyone's color if they are still in the system, along with their own."

WAIT{TIME = 0 MAX NORMAL(10.5,5/6)}; "all get to fence"
SEIZE{NAME = fence}; "jump over one at a time"
WAIT{TIME = 6}; "jump fence"
RELEASE{NAME = fence};
IF color = 0 THEN GOTO blacksheep;

"now only white sheep remain"
IF RANDOM > 0.5 THEN DEPART{}; "50% of white sheep excape"
GOTO slaughter; "tough luck kid"

blacksheep: IF RANDOM >= 0.2 THEN DEPART{}; "20% lucked out"

slaughter: mutton = mutton + 0 MAX NORMAL (65, 30/6); "add one sheep to mutton" DEPART{};

ARRIVE{TIME = EXPD(35)}; "born pigs"
dummy: WAIT{TIME = 0 MAX NORMAL(45,10/6)}; "run to and around fence"
IF RANDOM <= 0.1 THEN GOTO processpigs;
IF RANDOM <= 0.3 THEN GOTO dummy;
DEPART{};

processpigs: bacon = bacon + 0 MAX NORMAL(50,20); "makie backie"
DEPART{};

END.
Output for other quiz, but idea is the same.

MOR/DS 1.00

Date: 12/10/13  Time: 13:32:17

Y:\sheep.txt

MODEL DESCRIPTION

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

PROGRAM
"sheep and pigs"

DEFINITION

ATTRIBUTES = {color = 0}; "give each sheep their own color"
fence :RESOURCE = {};
mutton = 0;
bacon = 0;

LABELS = {blacksheep, dummy, processpigs, slaughter};

CONTROL

STOPTIME=8*60*60; "run 8 hours in seconds"
RANDOMIZE = ON;

LOGIC

[ 1]  [ 2] ARRIVE {TIME=UNIFORM(5,9)}; "born sheep"
[ 3] IF RANDOM <= 0.2 THEN color = 0 ELSE color = 1; "set 0 = black"
[ 4] WAIT{TIME = 0 MAX NORMAL(10.5,5/6)}; "get to fence"
[ 5] SEIZE{NAME = fence};
[ 6] WAIT{TIME = 0}; "jump fence"
[ 7] RELEASE{NAME = fence};
[ 8] IF color = 0 THEN GOTO blacksheep;
[ 9]
[10] "now only white sheep remain"
[11] IF RANDOM > 0.5 THEN DEPART{};
[12] GOTO slaughter; "tough luck kid"
[13]
[14] blacksheep: IF RANDOM >= 0.2 THEN DEPART{}; "you lucked out"
[15] slaughter: mutton = mutton + 0 MAX NORMAL (65, 30/6); "add one sheep to mutton"
[16] DEPART{};
[17]
[18] ARRIVE{TIME = EXPD(35)}; "born pigs"
[19] dummy: WAIT{TIME = 0 MAX NORMAL(45,10/6)}; "run to and around fence"
[20] IF RANDOM <= 0.1 THEN GOTO processpigs;
[21] IF RANDOM <= 0.3 THEN GOTO dummy;
[22] DEPART{};
processpigs: bacon = bacon + 0 MAX NORMAL(50,20); "makie backie"

DEPART();

END.

Simulation Clock : 28000.00

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>ARRIVE</td>
<td>4110</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>IF</td>
<td>4109</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>ASSIGNMENT</td>
<td>807</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>ASSIGNMENT</td>
<td>3302</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>WAIT</td>
<td>4109</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>SEIZE</td>
<td>4108</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>WAIT</td>
<td>4108</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>RELEASE</td>
<td>4107</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>IF</td>
<td>4107</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>GOTO</td>
<td>807</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>IF</td>
<td>3300</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>DEPART</td>
<td>1656</td>
<td>1656</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>GOTO</td>
<td>1644</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>BLACKSHEEP</td>
<td>IF</td>
<td>807</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>DEPART</td>
<td>652</td>
<td>652</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>SLAUGHTER</td>
<td>ASSIGNMENT</td>
<td>1799</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>DEPART</td>
<td>1799</td>
<td>1799</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>ARRIVE</td>
<td>834</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>DUMMY</td>
<td>WAIT</td>
<td>1148</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>IF</td>
<td>1144</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>GOTO</td>
<td>120</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>21</td>
<td>IF</td>
<td>1024</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>GOTO</td>
<td>315</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>22</td>
<td>DEPART</td>
<td>789</td>
<td>789</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>24</td>
<td>PROCESSPIG</td>
<td>ASSIGNMENT</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>25</td>
<td>DEPART</td>
<td>120</td>
<td>120</td>
<td></td>
</tr>
</tbody>
</table>

Resources

FENCE
Capacity = 1
Total Number = 4108
Minimum Capacity = 1
Maximum Capacity = Inf
Current Units = 1
Maximum Entities = 1

Maximum Units = 1
Average Util. = 0.570
Average Units = 0.57
Std Dev Units = 0.50
Average Time = 4.00

Default Queue Statistics
Capacity = Inf
Maximum Number = 1
Current Number = 0
Maximum Number = 129
Average Number = 0.00
Std Dev Number = 0.05
Average Time = 0.60

Global Variables
MUTTON = 117071.87

Page 2 of 3
BACON = 5969.75
Problem 3) (50 points) Our company runs a helicopter fire service for the U.S. Forest Service. We supply helicopters throughout the park. The purpose of this simulation is to determine if our proposed helicopter coverage is likely to be adequate. Run the simulation for a one month period, starting at midnight. Extra paper is attached for working this problem.

The Helicopter

Each helicopter covers a block in the shape of a rectangle approximately \( x = 40 \) miles wide by \( y = 30 \) miles deep. At the center of each block we construct a base for the helicopter and its support facilities (hanger, living quarters, water supply pond, etc.). Once called out, the helicopter takes 5 minutes to warm up, hitch up the bucket of water, and get the crew on board. Once over the water pond on the base it takes the helicopter 10 minutes to lower and fill the bucket. It then travels to the fire. Its average speed is 100 mph transporting the 400 gallon water bucket shown.

The Bucket

The bucket carries 400 gallons of water. It is empty at the support facility so the helicopter must fill it at the start of a mission. Partial bucket drops are not possible; it’s all or none. Once the water is dropped, it lowers the fire intensity by up to 200 Therms, depending on the efficiency of the drop. Drop efficiency = \( 1/ (1+(0.02*\text{wind velocity})^0.45) \). It takes no time to drop the water since it is done at full speed. If the Therm level of the fire lowered to, or below zero, it is extinguished and the helicopter returns to base. If not, it returns to the nearest pond to refill the bucket.

Water Supply

In the block you are studying, relative to the center of the block there is a second pond at \( x = 10 \) miles, \( y = 12 \) miles.

The Weather

The helicopter can fly in winds normally encountered in the area, but wind scatters the water and reduces the efficiency of the drop. Wind velocity in the area is 0 mph from midnight for 6 hours, 15 mph for the next 6 hours, 25 mph for the next 6 hours, and 5 mph until midnight.
The Fires

The fires are caused by lightning strikes, uniformly distributed in space across the block, i.e. it is uniformly likely to hit anywhere in the block. Strikes occur in time exponentially distributed with a mean of 20 hours. Before we can detect a fire it will have already reached an intensity of 300 Therms, after which they increase in intensity at a rate of two therms/minute. If a second fire starts in your block, it will be assigned to a helicopter from an adjoining block. Those fires starting after you have returned to base will be your responsibility.

Statistics

For a one month period, we need the following information:

- a count of how many fires you put out
- how many times outside help was required to put out second fires
- how much total water was used from the ponds
- how many hours your helicopter flew
- the highest therm level recorded for any fire
PROGRAM
   "put out fires"

DEFINITION

   eff = 1.0; "set efficiency at 100%"
coptertime = 0.0; "copter running time"
firesburning = 0; "number of fires currently burning"
totalfires = 0; "total number of fires experienced"
totalwater = 0; "total water taken from ponds"
xfire = 0;
yfire = 0;
xpond = 10; "pond location"
ypond = 12;
callforhelp = 0;
therms = 0;
maxtherms = 3; "keep track of hottest fire"
basetofire = 0; "distance from base to fire"
firetopond = 0; "distance from fire to pond"
dtowater = 0; "closest distance to water"
speed = 100/50; "speed of copter in miles per minute"
starttime = 0;

LABELS = {cat, fuel, mouse, dog, morefire, base, fire, nofire, dropwater, finishmission, quitting};

CONTROL

STOPTIME=30*24*60; "run a month in minutes"
RANDOMIZE = ON;

LOGIC

ARRIVE {TIME=0, LIMIT = 1}; "weather sets efficiency"
cat: eff = 1.0; "set efficiency = 100%"
   WAIT(TIME = 6*60); "wait 6 hours"
   eff = 1/(1+(0.02*15)^0.45); "reset efficiency"
   WAIT(TIME = 6*60); "wait 6 hours"
   eff = 1/(1+(0.02*25)^0.45); "reset efficiency"
   WAIT(TIME = 6*60); "wait 6 hours"
   eff = 1/(1+(0.02* 5)^0.45); "reset efficiency"
   WAIT(TIME = 6*60); "wait 6 hours"
  GOTO cat; "do it again"

ARRIVE {TIME=EXPD(20*60)}; "born fires"
   firesburning = firesburning + 1; "add 1 to number of fires currently burning"
   totalfires = totalfires + 1; "add 1 to number of fires experienced"
   IF firesburning >= 2 THEN GOTO dog ELSE GOTO mouse;
dog: callforhelp = callforhelp + 1; "add 1 to total calls for outside help"
firesburning = firesburning - 1; "2nd fire put out by others"
DEPART{};
"newly arrived fire leaves"
mouse: therms = 300; "fire is at 300 therms when we detect it"
xfire = CLNIFORM(-20,20); "get x coordinate of fire"
yfire = CLNIFORM(-15,15); "get y coordinate of fire"
morefire: WAIT{TIME = 1}; "wait a minute"
therms = therms + 2; "add therms to fire"
IF maxtherms < therms THEN maxtherms = therms; "keep track of hottest fire"
IF therms <= 0 THEN GOTO quitting; "fire extinguished"
GOTO morefire; "fire burns another minute"
quitting: firesburning = 0; "no fires burning"
DEPART{}

ARRIVE {TIME=0, LIMIT = 1}; "helicopter arrives"
base: IF firesburning = 0 THEN GOTO nofire ELSE GOTO fire;
nofire: WAIT{TIME = 10}; "wait ten minutes and check again"
GOTO base;

fire: starttime = CLOCKTIME; "set time helicopter starts up for this mission"
basetofire = (xfire^2+yfire^2)^0.5; "calculate distance from base to fire"
firetopond = ((xfire-xpond)^2 + (yfire-ypond)^2)^0.5; "calculate distance from fire to pond"
dtowater = basetofire;
from base to fire"
WAIT{TIME = 5}; "prepare and load copter"
WAIT{TIME = 10};
WAIT{TIME = dtowater/speed};
dropwater: totalwater = totalwater + 800;
from ponds"
therms = therms - 200*eff; "reduce therms by 200*efficiency"
IF therms <= 0 THEN GOTO finishmission; "fire extinguished, returning to base"
IF basetofire >= firetopond THEN dtowater = firetopond ELSE dtowater = basetofire;
"compute which water source is closer to the fire and set to dtowater"
WAIT{TIME = dtowater/speed};
WAIT{TIME = 10};
WAIT{TIME = dtowater/speed};
GOTO dropwater;

finishmission: WAIT{TIME = basetofire/speed}; "travel back to base"
coptertime = coptertime + CLOCKTIME - starttime; "calculate this mission time"
GOTO base; "back at base"

Please see the computer run and comments at the end of the run. It should really show you how extremely valuable simulation can be.

Questions you can answer and which should be asked:
Do I need a faster helicopter? Depends on how many times you had to ask for help, and if they were able to come.
Do I need a bigger bucket? Same as above and if your helicopter will carry it.
Will the ponds run dry? Check amount of water in them plus that added from expected rainfall, vs. how much you use a month.
What about running out of fuel? Simply add code to calculate burn rate from the tank and force a return to base when tanks get low.
How can I get the adjacent helicopters up and running? Simply copy and paste the entire logic block directly above your END statement and change all the names from eff to eff2, fuel to fuel2, etc. to represent the second complete helicopter. You will have to have your first helicopter call out the second one rather than it just DEPARTing as abivem and then let the second set of code do its thing. You can put many more helicopters in here like that and have each one, when necessary, call out the closest one not currently on a mission.
How can I make sure it is really running correctly?
One thing you can do is put PRINTTIME = 10: (or any number) to print out the answers on a file at every time CLOCKTIME >= multiples of 10.
MOR/DS 1.00

Date: 12/9/13 Time: 17:05:39
Y:\fire2.txt

MODEL DESCRIPTION

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

PROGRAM
"put out fires"

DEFINITION

eff = 1.0; "set efficiency at 100%"
coptertime = 0.0; "copter running time"
firesburning = 0; "number of fires currently burning"
totalfires = 0; "total number of fires experienced"
totalwater = 0; "total water taken from ponds"
xfire = 0;
yfire = 0;
xpond = 10; "pond location"
ypond = 12;
callforhelp = 0;
therms = 0;
maxtherms = 0; "keep track of hottest fire"
basetofire = 0; "distance from base to fire"
firetopond = 0; "distance from fire to pond"
dtowater = 0; "closest distance to water"
speed = 100/60; "speed of copter in miles per minute"
starttime = 0;

LABELS = {cat, fuel, mouse, dog, morefire, base, fire, nofire, drop
water, finishmission, quitting};

CONTROL
STOPTIME=30*24*60; "run a month in minutes"
RANDOMIZE = ON;

LOGIC
[ 1]
[ 2] ARRIVE {TIME=0, LIMIT = 1}; "weather sets efficiency"
[ 3] cat: eff = 1.0; "set efficiency = 100%"
[ 4] WAIT{TIME = 6*60}; "wait 6 hours"
[ 5] eff = 1/(1+(0.02*15)*0.45); "reset efficiency"
[ 6] WAIT{TIME = 6*60}; "wait 6 hours"
[ 7] eff = 1/(1+(0.02*25)*0.45); "reset efficiency"
[ 8] WAIT{TIME = 6*60}; "wait 6 hours"
eff = 1/(1+(0.02*5)^0.45); "reset efficiency"

WAIT(TIME = 6*60); "wait 6 hours"

GOTO cat; "do it again"

ARRIVE {TIME=EXPD(20*60)}; "born fires"
firesburning = firesburning + 1; "add 1 to number of fires currently burning"
totalfires = totalfires + 1; "add 1 to number of fires experienced"

IF firesburning >= 2 THEN GOTO dog ELSE GOTO mouse;

dog: callforhelp = callforhelp + 1; "add 1 to total calls for outside help"
firesburning = firesburning - 1; "2nd fire put out by others"
DEPART(); "newly arrived fire leaves"

mouse: therms = 300; "fire is at 300 therms when we detect it"
xfire = CUNIFORM(-20,20); "get x coordinate of fire"
yfire = CUNIFORM(-15,15); "get y coordinate of fire"
morfire: WAIT{TIME = 1}; "wait a minute"
therms = therms + 2; "add therms to fire"
IF maxtherms < therms THEN maxtherms = therms; "keep track of hottest fire"

IF therms <= 0 THEN GOTO quitting; "fire extinguished"
GOTO morefire; "fire burns another minute"

quitting: firesburning = 0; "no fires burning"
DEPART();

ARRIVE {TIME=0, LIMIT = 1}; "helicopter arrives"

base: IF firesburning = 0 THEN GOTO nofire ELSE GOTO fire;

nofire: WAIT{TIME = 10}; "wait ten minutes and check again"

GOTO base;

fire: starttime = CLOCKTIME; "set time helicopter starts up for this mission"
basetofire = (xfire^2 + yfire^2)^0.5; "calculate distance from base to fire"
firetopond = ((xfire-xpond)^2 + (yfire-ypond)^2)^0.5; "calculate distance from fire to pond"
dtowater = basetofire; "first trip always goes from base to fire"

WAIT{TIME = 5}; "prepare and load copter"

WAIT{TIME = 10}; "take off and get water"

WAIT{TIME = dtowater/speed}; "time to get to fire"

donewater: totalwater = totalwater + 800; "calculate total water from ponds"

therms = therms - 200*eff; "reduce therms by
200*efficiency"
[ 50] IF therm <= 0 THEN GOTO finishmission; "fire
extinguished, returning to base"
[ 51] IF basetofire >= firetopond THEN dtowater = firetopond ELSE
dtowater = basetofire;
[ 52] "compute which water source is closer to the fire and
set to dtowater"
[ 53] [ 54] WAIT(TIME = dtowater/speed); "go to water
source"
[ 55] WAIT(TIME = 10); "get water"
[ 56] WAIT(TIME = dtowater/speed);
[ 57] GOTO dropwater;
[ 58] [ 59] finishmission: WAIT(TIME = basetofire/speed); "travel back to
base"
[ 60] coptertime = coptertime + CLOCKTIME - starttime; "calculate
this mission time"
[ 61] GOTO base; "back at base"
[ 62]
[ 63] END.

Simulation Clock : 43200.00

<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>ARRIVE</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>CAT</td>
<td>ASSIGNMENT</td>
<td>31</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td>WAIT</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td>ASSIGNMENT</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td>WAIT</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td></td>
<td>ASSIGNMENT</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td></td>
<td>WAIT</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td></td>
<td>ASSIGNMENT</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
<td>WAIT</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td></td>
<td>GOTO</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td></td>
<td>ARRIVE</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td></td>
<td>ASSIGNMENT</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>17</td>
<td></td>
<td>ASSIGNMENT</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>18</td>
<td></td>
<td>IF</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td></td>
<td>GOTO</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>18</td>
<td></td>
<td>GOTO</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>DOG</td>
<td>ASSIGNMENT</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td></td>
<td>ASSIGNMENT</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>22</td>
<td></td>
<td>DEPART</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>MOUSE</td>
<td>ASSIGNMENT</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>25</td>
<td></td>
<td>ASSIGNMENT</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>26</td>
<td></td>
<td>ASSIGNMENT</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>27</td>
<td>MOREFIRE</td>
<td>WAIT</td>
<td>3526</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td></td>
<td>ASSIGNMENT</td>
<td>3526</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>29</td>
<td></td>
<td>IF</td>
<td>3526</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>29</td>
<td></td>
<td>ASSIGNMENT</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>30</td>
<td></td>
<td>IF</td>
<td>3526</td>
<td>0</td>
</tr>
</tbody>
</table>
This run shows you had 42 fires in your area, of which you had to have help on 5, and thus put out 37 of them on your own. The maximum Therms of any fire was 374. The copter flew 3629 hours so you can tell when scheduled maintenance will be required.
READ THE FOLLOWING GENERAL EXAMINATION RULES:

1) Do not put your completed work anywhere that it can be seen. If any part of your work can be seen by others it will be confiscated and you will not be permitted to rework those problems. Place any pages of your work face down on your desk under your existing work, not on the floor next to you where it is visible.
2) Please remove your hat. If it is part of your head, turn it around backwards.
3) If your work is not legible, or if I cannot follow your logic at a glance, it will receive no credit. This paper must be written to acceptable engineering standards for credit. Please take this seriously as it will affect your grade.
4) You may work on the front or back of this paper. Just note if work is on the back.
5) You can use your own paper or paper supplied at the front of the room.
6) You MUST specify what you are doing every step of the way. ONLY if I can follow what you are trying to do and where you are getting your numbers from, will you receive partial credit should you go off track.
7) Write big and use lots of paper, leaving me room to grade your paper. If there is no room to tell you why points were deducted, I will only show you the point deduction and let you try and figure out why.

I have read and understand all of the above instructions: ____________ (Initials)

Ethical Standards:

Upon accepting admission to Texas A&M University, a student immediately assumes a commitment to uphold the Honor Code, to accept responsibility for learning, and to follow the philosophy and rules of the Honor System. Students will be required to state their commitment on examinations, research papers, and other academic work. Ignorance of the rules does not exclude any member of the TAMU community from the requirements or the processes of the Honor System.

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this exam."

__________________________________
Signature of student

Please do not open this exam until you are told to do so.
Problem 1) (25 points) Last night I had the strangest dream. I dreamed that sheep entered through a door in a barn according to a uniform distribution with a minimum time of 5 and a maximum of 9 seconds. 25% of the sheep were black, the rest were white. After entering, the sheep ran over to a fence. It took them from 8 to 13 seconds to get to it, distributed normally. They then proceeded to jump the fence which took 4 seconds. Once over the fence, 20% of the black sheep and 50% of the white sheep were turned into mutton. The rest ran out the back door and escaped. Each sheep processed yielded between 50 and 80 pounds of mutton, distributed normally.

Pigs also entered the barn, exponentially distributed at a mean of 35 seconds. They were all pink. They also ran up to the fence, but ran around it, taking between 40 and 50 seconds distributed normally. Once past the fence, 10% of the pigs were caught and turned into bacon, and the rest ran towards the exit. However, not being very smart, 30% of those returned to the gate and went through the system in the same manner as before. The rest went escaped. Each pig yielded a mean of 50 pounds of bacon with a sigma of 20 pounds.

Write a BOSS program to determine how much bacon and how much mutton I had after 8 hours of dreaming.
PROGRAM  
"sheep and pigs"

DEFINITION

ATTRIBUTES = {color = 0}; "give each sheep their own color"
  fence :RESOURCE = {};
  mutton = 0;
  bacon = 0;

LABELS = {blacksheep, dummy, processpigs, slaughter};

CONTROL

STOPTIME = 8*60*60; "run 8 hours in seconds"
RANDOMIZE = ON;

LOGIC

ARRIVE {TIME=CUNIFORM(5,9)}; "born black and white sheep"
IF RANDOM <= 0.2 THEN color = 0 ELSE color = 1; "set 0 = black"

"Note that you MUST use an ATTRIBUTE if you set their colors here, because otherwise the next sheep coming shortly thereafter will change everyone's color if they are still in the system, along with their own."

WAIT{TIME = 0 MAX NORMAL(10.5, 5/6)}; "all get to fence"
SEIZE{NAME = fence}; "jump over one at a time"
WAIT{TIME = 4}; "jump fence"
RELEASE{NAME = fence};
IF color = 0 THEN GOTO blacksheep;

"now only white sheep remain"
IF RANDOM > 0.5 THEN DEPART{}; "50% of white sheep escape"
GOTO slaughter; "tough luck kid"

blacksheep: IF RANDOM >= 0.2 THEN DEPART{}; "20% lucked out"

slaughter: mutton = mutton + 0 MAX NORMAL (65, 30/6); "add one sheep to mutton"
DEPART{};

ARRIVE{TIME = EXPD(35)}; "born pigs"
dummy: WAIT{TIME = 0 MAX NORMAL(45, 10/6)}; "run to and around fence"
IF RANDOM <= 0.1 THEN GOTO processpigs;
IF RANDOM <= 0.3 THEN GOTO dummy;
DEPART{};

processpigs: bacon = bacon + 0 MAX NORMAL(50, 20); "make backie"
DEPART{};

END.
Problem 2) (25 points) My bank is a really nice bank. Today is November 30\textsuperscript{th}, 2013. Normally if I miss a payment, they charge me a really hefty late penalty. However, they just emailed me and said that since it was Christmas, and since they were really nice guys, I could skip my upcoming end of December payment without a late charge. Having maxed out the card at $20,000, I had already decided to cut it up and stop charging things, since they were charging me 24\% interest. I had intended pay off my balance in the next 24 months in equal installments, but due to their offer I think I might go ahead skip the December 31\textsuperscript{st} payment, and then go ahead and pay them off in the following 24 months. How much extra interest will I end up paying if I decide to skip my December payment?

\[ A_1 = \frac{P}{18.9139} = 20 \times 0.9139 = 1.05742 \text{$/month} \]

Interest = Total payout - $P = 1.05742(24) - 20 = 5.37813

\[ A_2 = \frac{20}{(18.9139)(0.9804)} = 1.07857 \text{$/month} \]

Interest = (1.07857$/month)(24 months) - 20

= 5.88572

\[ \Delta \text{interest} = 5.886^k - 5.378^k = 508 \text{ more in interest} \]
Problem 3) (50 points) Our company runs a helicopter fire service for the U.S. Forest Service. We supply helicopters throughout the park. The purpose of this simulation is to determine if our proposed helicopter coverage is likely to be adequate. Run the simulation for a one month period, starting at midnight. Extra paper is attached for working this problem.

The Helicopter

Each helicopter covers a block in the shape of a rectangle approximately \( x = 50 \) miles wide by \( y = 40 \) miles deep. At the center of each block we construct a base for the helicopter and its support facilities (hanger, living quarters, water supply pond, etc.). Once called out, the helicopter takes 5 minutes to warm up, hitch up the bucket of water, and get the crew on board. Once over the water pond on the base it takes the helicopter 10 minutes to lower and fill the bucket. It then travels to the fire. Its average speed is 100 mph transporting the 400 gallon water bucket shown.

The Bucket

The bucket carries 400 gallons of water. It is empty at the support facility so the helicopter must fill it at the start of a mission. Partial bucket drops are not possible; it's all or none. Once the water is dropped, it lowers the fire intensity by up to 200 Therms, depending on the efficiency of the drop. Drop efficiency \( = 1/(1+(0.02 \times \text{wind velocity})^0.45) \). It takes no time to drop the water since it is done at full speed. If the Therm level of the fire lowered to, or below zero, it is extinguished and the helicopter returns to base. If not, it returns to the nearest pond to refill the bucket.

Water Supply

In the block you are studying, relative to the center of the block there is a second pond at \( x = 10 \) miles, \( y = 12 \) miles.

The Weather

The helicopter can fly in winds normally encountered in the area, but wind scatters the water and reduces the efficiency of the drop. Wind velocity in the area is 0 mph from midnight for 6 hours, 15 mph for the next 6 hours, 25 mph for the next 6 hours, and 5 mph until midnight.
The Fires

The fires are caused by lightning strikes, uniformly distributed in space across the block, i.e. it is uniformly likely to hit anywhere in the block. Strikes occur in time exponentially distributed with a mean of 20 hours. Before we can detect a fire it will have already reached an intensity of 300 Therms, after which they increase in intensity at a rate of two therms/minute. If a second fire starts in your block, it will be assigned to a helicopter from an adjoining block. Those fires starting after you have returned to base will be your responsibility.

Statistics

For a one month period, we need the following information:

- a count of how many fires you put out
- how many times outside help was required to put out second fires
- how much total water was used from the ponds
- how many hours your helicopter flew
- the highest therm level recorded for any fire

Same as other quiz