Department of Mathematical Sciences
Carnegie Mellon University

21-393 Operations Research II
Test 1

Name:____________________________________

Problem	Points	Score
1	40	
2	40	
3	20	
Total	100	

Q1: (40pts)
(a) Solve the following knapsack problem, writing the results of the dynamic programming recursion in a table. You will not score any points for just writing down the answer:

$$\begin{array}{lrcrcrcl}
\text{maximise} & 3x_1 & + & 7x_2 & + & 15x_3 & & \\
\text{subject to} & & & & & & & \\
 & 2x_1 & + & 3x_2 & + & 6x_3 & \leq & 10
\end{array}$$

$$x_1, x_2, x_3 \geq 0 \text{ and integer.}$$

Your answer should consist of a table.
(b) Using the answer to part (a), solve the following problem:

$$\begin{array}{lrcrcrcl}
\text{minimise} & 2x_1 & + & 3x_2 & + & 6x_3 & & \\
\text{subject to} & & & & & & & \\
 & 3x_1 & + & 7x_2 & + & 15x_3 & \geq & 20
\end{array}$$

$$x_1, x_2, x_3 \geq 0 \text{ and integer.}$$

(This does not require any new computations!)

Q2: (40pts) A system can be in 3 states 1,2,3 and the cost of moving from state i to state j in one period is $c(i,j)$, where the $c(i,j)$ are given in the matrix below. The one period discount factor α is 1/2.
The aim is to find a policy which simultaneously minimises the discounted cost of operating from any starting state. Start with the policy

$$\pi(1) = 1, \pi(2) = 3, \pi(3) = 2.$$

Evaluate this policy. Is it optimal? If not find an improved policy.
YOU DO NOT NEED TO EVALUATE THIS NEW POLICY OR FIND AN OPTIMAL STRATEGY.
The matrix of costs is

$$\begin{bmatrix} 8 & 3 & 1 \\ 4 & 2 & 8 \\ 1 & 8 & 2 \end{bmatrix}$$

Q3: (20pts) You are given a set of n types of rectangular 3-D boxes, where the ith box has height $h(i)$, width $w(i)$ and depth $d(i)$ (all real numbers). You want to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base. It is also allowable to use multiple instances of the same type of box.
Formulate a Dynamic Programming recursion that can be used to solve this problem.