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:

$$\begin{array}{lccccccc} \text{maximise} & 3x_1 & + & 7x_2 & + & 17x_3 & & \\ \text{subject to} & & & & & & & \\ & 2x_1 & + & 3x_2 & + & 7x_3 & \leq & 10 \end{array}$$

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

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

$$\begin{array}{lccccccc} \text{minimise} & 2x_1 & + & 3x_2 & + & 7x_3 & & \\ \text{subject to} & & & & & & & \\ & 3x_1 & + & 7x_2 & + & 17x_3 & \geq & 20 \end{array}$$

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

(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

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

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

$$\begin{bmatrix} 5 & 10 & 1 \\ 8 & 2 & 2 \\ 1 & 10 & 2 \end{bmatrix}$$

Q3: (20pts) I, J are intervals of length n. Every pair of sub-intervals $I' \subseteq I, J' \subseteq J$ is given a value $v(I', J')$. Here we are restricting our attention to intervals that have integer endpoints. Give a Dynamic Programming algorithm for partitioning I into consecutive intervals $I_1, I_2, \ldots, I_m$ and J into consecutive intervals $J_1, J_2, \ldots, J_m$ in order to maximise the total value $v(I_1, J_1) + v(I_2, J_2) + \cdots + v(I_m, J_m)$. The intervals chosen must be such that $I_t \cap J_t \neq \emptyset$ for $t = 1, 2, \ldots, m$. There is no restriction on m.