Travis Robinson
Ben Henning
Scott Williams
CS325
Project 3

Problem 1:
A.i
p_iw_j = Route from Plant i to Warehouse j
w_jr_k = Route from Warehouse j to Retailer k

Z = min
$(10p_1w_1+15p_1w_2+11p_2w_1+8p_2w_2+13p_3w_1+8p_3w_2+9p_3w_3+14p_4w_2+8p_4w_3+5w_1r_1+6w_1r_2+7w_1r_3+10w_1r_4$
$+12w_2r_3+8w_2r_4+10w_2r_5+14w_2r_6+14w_3r_4+12w_3r_5+12w_3r_6+6w_3r_7)$

Constraints:
$p_1w_1+p_1w_2$ <= 150 //constraint of Plant 1 production
$p_2w_1+p_2w_2$ <= 450 //constraint of Plant 2 production
$p_3w_1+p_3w_2+p_3w_3$ <= 250 //constraint of Plant 3 production
$p_4w_2+p_4w_3$ <= 150 //constraint of Plant 4 production
w_1r_1 >= 100 //constraint of what retailer 1 requires
w_1r_2 >= 150 //constraint of what retailer 2 requires
$w_1r_3+w_2r_3$ >= 100 //constraint of what retailer 3 requires
$w_1r_4+w_2r_4$+w3r4 >= 200 //constraint of what retailer 4 requires
$w_2r_5+w_3r_5$ >= 200 //constraint of what retailer 5 requires
$w_2r_6+w_3r_6$ >= 150 //constraint of what retailer 6 requires
w_3r_7 >= 100 //constraint of what retailer 7 requires
$p_1w_1+p_2w_1+p_3w_1-w_1r_1-w_1r_2-w_1r_3-w_1r_4$ >= 0 //production must be greater than retailer requirement
$p_1w_2+p_2w_2+p_3w_2+p_4w_2-w_2r_3-w_2r_4-w_2r_5-w_2r_6$ >= 0 //production must be greater than retailer requirement
$p_3w_3+p_4w_3-w_3r_4-w_3r_5-w_3r_6-w_3r_7$ >= 0 //production must be greater than retailer requirement
p_1w_1 >= 0 //non-negativity constraint
p_1w_2 >= 0 //non-negativity constraint
p_2w_1 >= 0 //non-negativity constraint
p_2w_2 >= 0 //non-negativity constraint
p_3w_1 >= 0 //non-negativity constraint
p_3w_2 >= 0 //non-negativity constraint
p_3w_3 >= 0 //non-negativity constraint
p_4w_2 >= 0 //non-negativity constraint
p_4w_3 >= 0 //non-negativity constraint
w_1r_1 >= 0 //non-negativity constraint
w_1r_2 >= 0 //non-negativity constraint
w_1r_3 >= 0 //non-negativity constraint
w_1r_4 >= 0 //non-negativity constraint
w_2r_3 >= 0 //non-negativity constraint
w_2r_4 >= 0 //non-negativity constraint
w_2r_5 >= 0 //non-negativity constraint
w_2r_6 >= 0 //non-negativity constraint

w_3r_4 >= 0 //non-negativity constraint
w_3r_5 >= 0 //non-negativity constraint
w_3r_6 >= 0 //non-negativity constraint
w_3r_7 >= 0 //non-negativity constraint

A.ii
```
MIN
10p1w1+15p1w2+11p2w1+8p2w2+13p3w1+8p3w2+9p3w3+14p4w2+8p4w3+5w1r1+6w1r2+7w1r3+10w1r4+12w2r3+8w2r4+10w2r5+14w2r6+14w3r4+12w3r5+12w3r6+6w3r7
ST
p1w1+p1w2 < 150
p2w1+p2w2 < 450
p3w1+p3w2+p3w3 < 250
p4w2+p4w3 < 150
w1r1 > 100
w1r2 > 150
w1r3+w2r3 > 100
w1r4+w2r4+w3r4 > 200
w2r5+w3r5 > 200
w2r6+w3r6 > 150
w3r7 > 100
p1w1+p2w1+p3w1-w1r1-w1r2-w1r3-w1r4>0
p1w2+p2w2+p3w2+p4w2-w2r3-w2r4-w2r5-w2r6>0
p3w3+p4w3-w3r4-w3r5-w3r6-w3r7>0
p1w1 > 0
p1w2 > 0
p2w1 > 0
p2w2 > 0
p3w1 > 0
p3w2 > 0
p3w3 > 0
p4w2 > 0
p4w3 > 0
w1r1 > 0
w1r2 > 0
w1r3 > 0
w1r4 > 0
w2r3 > 0
w2r4 > 0
w2r5 > 0
w2r6 > 0
w3r4 > 0
w3r5 > 0
w3r6 > 0
w3r7 > 0
END
```

A.iii.

Minimum Shipping Cost: 17100.00
Optimal Shipping Routes:
P1W1 150.000000
P1W2 0.000000
P2W1 200.000000
P2W2 250.000000
P3W1 0.000000
P3W2 150.000000
P3W3 100.000000
P4W2 0.000000
P4W3 150.000000
W1R1 100.000000
W1R2 150.000000
W1R3 100.000000
W1R4 0.000000
W2R3 0.000000
W2R4 200.000000
W2R5 200.000000
W2R6 0.000000
W3R4 0.000000
W3R5 0.000000
W3R6 150.000000
W3R7 100.000000

B. It is not feasible to ship all the refrigerators without warehouse 2 because that leaves warehouse 3 as the only warehouse able to ship to retailers 5, 6, and 7, and plants 3 and 4 are unable to provide for the demand given that they can only ship to warehouse 3, leaving a shortage of 50 refrigerators for those three retailers. Because this is not feasible, there is no optimal solution.

C. This is feasible because it addresses the issue of not being able to get sufficient refrigerators to the retailers 5, 6, and 7 because now warehouse 3 is not the only one supplying those warehouses. The optimal solution in this case is 18300.00
P1W1 150.000000
P1W2 0.000000
P2W1 350.000000
P2W2 100.000000
P3W1 0.000000
P3W2 0.000000
P3W3 250.000000
P4W2 0.000000
P4W3 150.000000
W1R1 100.000000
W1R2 150.000000
W1R3 100.000000
W1R4 150.000000
W2R3 0.000000
W2R4 50.000000

W2R5 50.000000
W2R6 0.000000
W3R4 0.000000
W3R5 150.000000
W3R6 150.000000
W3R7 100.000000

And in fact, if management wished, they could limit warehouse 2 to 50 refrigerators, though this may give an even less optimal solution.

D.
x_{ij} = cost of shipping from plant i to warehouse j
x_{jk} = cost of shipping from warehouse j to retailer k
$p_i w_j$ = the number of units from plant i to warehouse j
$w_j r_k$ = the number of units from warehouse k to plant k
$Z = \min (x_{ij} \Sigma_{i=1 \text{ to } i} \Sigma_{i=1 \text{ to } j} p_i w_j + x_{jk} \Sigma_{i=1 \text{ to } j} \Sigma_{i=1 \text{ to } k} w_j r_k)$

Constraints:
$\Sigma_{i=1 \text{ to } i} \Sigma_{i=1 \text{ to } j} p_i w_j <= p_i$, p_i = production of plant i
$\Sigma_{i=1 \text{ to } j} \Sigma_{i=1 \text{ to } k} w_j r_k >= r_k$, r_k = retailer demand
$\Sigma_{i=1 \text{ to } i} p_i >= \Sigma_{i=1 \text{ to } k} r_k$, supply must be greater than or equal to retailer demand
$p_i w_j >= 0$, plants can't produce negative product, can't ship negative product
$w_j r_k >= 0$, retailers can't sell negative amount of product, warehouses can't ship negative amount of product

Problem 2:
Tomato = T, Lettuce = L, Spinach = S, Carrot = C, Sunflower Seeds = SS, Smoked Tofu = ST, Chickpeas = CP, Oil = O

Part A:
i)

Minimize kcal = 21T + 16L + 40S + 41C + 585SS + 120ST + 164CP + 884O

Subject to:

.85T + 1.62L + 2.86S + .93C + 23.4SS + 16ST + 9CP + 0O >= 15
.33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O >= 2
.33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O <= 8
4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP + 0O >= 4
9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP + 0O <= 200
L + S >= 0.4(T + L + S + C + SS + ST + CP + O)
T, L, S, C, SS, ST, CP, O >= 0

ii) Using Lindo:
Code:
```
MIN 21T + 16L + 40S + 41C + 585SS + 120ST + 164CP + 884O
ST
        .85T + 1.62L + 2.86S + .93C + 23.4SS + 16ST + 9CP + 0O > 15
        .33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O > 2
        .33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O < 8
        4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP + 0O > 4
        9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP + 0O < 200
        L + S - .4T - .4L - .4S  - .4C - .4SS - .4ST - .4CP - .4O > 0
        T > 0
        L > 0
        S > 0
        C > 0
        SS > 0
        ST > 0
        CP > 0
        O > 0
END
```

The optimal solution is:
0.585480 * 100 = 58.548 grams of Lettuce
0.878220 * 100 = 87.822 grams of Smoked Tofu
Total calories: 114.7541

iii) Cost = 58.548g * $1.00 / 100g + 87.822g * $2.15 / 100g = $2.473653, or **$2.47**

Part B:
i)
Minimize cost = 1T + .75L + .5S + .5C + .45SS + 2.15ST + .95CP + 2O

Subject to:

.85T + 1.62L + 2.86S + .93C + 23.4SS + 16ST + 9CP + 0O >= 15
.33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O >= 2
.33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O <= 8
4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP + 0O >= 4
9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP + 0O <= 200
L + S >= 0.4(T + L + S + C + SS + ST + CP + O)
T, L, S, C, SS, ST, CP, O >= 0

ii) Using Lindo:
Code:

```
MIN 1T + .75L + .5S + .5C + .45SS + 2.15ST + .95CP + 2O
ST
        .85T + 1.62L + 2.86S + .93C + 23.4SS + 16ST + 9CP + 0O > 15
        .33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O > 2
        .33T + .2L + .39S + .24C + 48.7SS + 5ST + 2.6CP + 100O < 8
        4.64T + 2.37L + 3.63S + 9.58C + 15SS + 3ST + 27CP + 0O > 4
        9T + 28L + 65S + 69C + 3.8SS + 120ST + 78CP + 0O < 200
        L + S - .4T - .4L - .4S  - .4C - .4SS - .4ST - .4CP - .4O > 0
        T > 0
        L > 0
        S > 0
        C > 0
        SS > 0
        ST > 0
        CP > 0
        O > 0
END
```

Optimal Solution:
Spinach: 83.2298 grams
Sunflower Seeds: 9.6083 grams
Chickpeas: 115.2364 grams
Cost = $1.554133, or $1.55

iii)
Calories = 83.2298g * 40 / 100g + 9.6083g * 585 / 100g + 115.2364g * 164 / 100g =
278.488171 kcal

Part C:
i)
One possible way would be to add a constraint that the total calories should be under 250 and to minimize the cost. Another way would be to add a constraint that the cost must be less than or equal to $2.00 and then to minimize the calories.

ii)
Spinach: 76.1996g
Sunflower Seeds: 9.3830g
Smoked Tofu: 16.8941g
Chickpeas: 88.0222g
Cost: 1.622657, or $1.62
Calories: 249.999718 kcal

iii)
Since it wasn't specified whether Veronica could sell more salads the further below 250 the calorie count got, it seemed more reasonable to set a constraint that the calories should not exceed 250 and then try to get the cost as low as possible, since it has a more obvious benefit of increasing the profit per salad. This was done by using the Lindo code from part B and adding the following constraint:

$21T + 16L + 40S + 41C + 585SS + 120ST + 164CP + 884O < 250$

However, in the event that lowering the calorie count vastly increases the amount of sales, this would not be the best solution. Without knowing how much priority to give reducing cost vs. reducing calories, it seems like any combination of ingredients that keeps the cost less than or equal to $2.00 and the calorie count less than or equal to 250 could be considered equally "optimal."

Problem 3

a) What are the lengths of the shortest paths from vertex a to all other vertices.

I chose to use Excel to solve this problem. I listed each source and destination paired with the distance from the source to the destination. I used a supply and demand listing for the list of each node to indicate the source and destination (1 for source and -1 for destination). Then used a reference column for the flow that used the formula:

> SUMIF([Source column], [current node], [list of intersected vertices]) -
> SUMIF([Destination column], [current node], [list of intersected vertices])

to determine the net flow of each edge. I then used the Data Solver to minimize the list of intersected vertices with the exception that the “Net Flow” column must equal the “Supply and Demand” column, mimicking a matrix that indicated the current two vertices. Here is a picture of the spreadsheet.

=SUMPRODUCT(OnRoute,D3:D39)

Source	Dest	Distance	On Route	Nodes	Net Flow		Supp/Dem
A	B	2	1	A	1	=	1
A	C	3	0	B	0	=	0
A	D	8	0	C	0	=	0
A	H	9	0	D	0	=	0
B	A	4	0	E	0	=	0
B	C	5	0	F	0	=	0
B	E	7	0	G	0	=	0
B	F	4	1	H	0	=	0
C	D	10	0	I	0	=	0
C	B	5	0	J	0	=	0
C	H	9	0	K	0	=	0
C	I	11	0	L	0	=	0
C	F	4	0	M	-1	=	-1
D	A	8	0				
D	G	2	0				
D	J	5	0				
D	F	1	0	Total	17		
E	H	5	0				
E	C	4	0				
E	I	10	0				
F	I	2	1				
F	G	2	0				
G	D	2	0				
G	J	8	0				
G	K	12	0				
H	I	5	0				
H	K	10	0				
I	A	20	0				
I	K	6	0				
I	J	2	1				
I	M	12	0				
J	I	2	0				
J	K	4	0				
J	L	5	1				
K	H	10	0				
K	M	10	0				
L	M	2	1				

And the list of the shortest path from A to each vertex:

A to B: {AB} = 2

A to C: {AC} = 3

A to D: {AD} = 8

A to E: {AB, BE} = 9

A to F: {AB, BF} = 6

A to G: {AB, BF, FG} = 8
A to H: {AH} = 9
A to I: {AB, BF, FI} = 8
A to J: {AB, BF, FI, IJ} = 10
A to K: {AB, BF, FI, IK} = 14
A to L: {AB, BF, FI, IJ, JL} = 15
A to M: {AB, BF, FI, IJ, JL, LM} = 17

b) If a vertex z is added to the graph for which there is no path from vertex a to vertex z, what will be the result when you attempt to find the lengths of shortest paths as in part a).

The only possible ways for vertex *z* to be added to the graph that does not have a path from *a* to *z* is if *z* is not connected to *any* of the current vertices or has the wrong direction. In this case, there would be no way to set up the problem as there would either be no distance from *z* to any other vertex and the program would not be able to solve for anything, or the program would not be able to find any edges that lead to *z* because *z* is not listed as a possible destination.

c) What are the lengths of the shortest paths from each vertex to vertex m. How can you solve this problem with just one linear program?

Exactly the same as part a, except that my source is variable and the destination is always *m*.

A to M: {AB, BF, FI, IJ, JL, LM} = 17
B to M: {BF, FI, IJ, JL, LM} = 15
C to M: {CF, FI, IJ, JL, LM} = 15
D to M: {DJ, JL, LM} = 12
E to M: {EC, CF, FI, IJ, JL, LM} = 19
F to M: {FI, IJ, JL, LM} = 11
G to M: {GD, DF, FI, IJ, JL, LM} = 14
H to M: {HI, IJ, JL, LM} = 14
I to M: {IJ, JL, LM} = 9
J to M: {JL, LM} = 7
K to M: {KM} = 10
L to M: {LM} = 2

d) Suppose that all paths must pass through vertex i. How can you calculate the length of the shortest path from any vertex x to vertex y that pass through vertex i (for all x,y V)? Calculate the lengths of these paths for the given graph. (Note for some vertices x and y it may be impossible to pass through vertex i).

This problem would need to be split up into two problems, the first problem having the source be ***x*** and the destination be ***i***, the second would need to have the source be ***i*** and the destination be ***y***. This problem can be solved by understanding that every path from source ***x*** must have the component of the path from ***x*** to ***i*** plus the component of the path from ***i*** to ***y***. To simplify the problem, you can figure out the path from each source to ***i*** and from ***i*** to each destination and add the corresponding lengths together to get the final path instead of doing each calculation individually. There is no possible path from ***L*** to ***i*** or from ***m*** to ***i***, so those paths are not listed.

A to B: 30	A to C: 31	A to D: 36	A to E: 37	A to F: 34	A to G: 36
A to H: 24	A to I: 8	A to J: 10	A to K: 14	A to L: 15	A to M: 17
B to A: 26	B to C: 29	B to D: 34	B to E: 35	B to F: 34	B to G: 36
B to H: 22	B to I: 6	B to J: 8	B to K: 12	B to L: 13	B to M: 15
C to A: 26	C to B: 28	C to D: 34	C to E: 35	C to F: 32	C to G: 34
C to H: 22	C to I: 6	C to J: 8	C to K: 12	C to L: 13	C to M: 15
D to A: 23	D to B: 25	D to C: 26	D to E: 32	D to F: 29	D to G: 31
D to H: 19	D to I: 3	D to J: 5	D to K: 9	D to L: 10	D to M: 12
E to A: 30	E to B: 32	E to C: 33	E to D: 38	E to F: 36	E to G: 38
E to H: 26	E to I: 10	E to J: 12	E to K: 16	E to L: 17	E to M: 19
F to A: 22	F to B: 24	F to C: 25	F to D: 30	F to E: 31	F to G: 30
F to H: 18	F to I: 2	F to J: 4	F to K: 8	F to L: 9	F to M: 11
G to A: 25	G to B: 27	G to C: 28	G to D: 33	G to E: 34	G to F: 33
G to H: 21	G to I: 5	G to J: 7	G to K: 11	G to L: 12	G to M: 14
H to A: 25	H to B: 27	H to C: 28	H to D: 33	H to E: 34	H to F: 33
H to G: 33	H to I: 5	H to J: 7	H to K: 11	H to L: 12	H to M: 14
I to A: 20	I to B: 22	I to C: 23	I to D: 28	I to E: 29	I to F: 26
I to G: 28	I to H: 16	I to J: 2	I to K: 6	I to L: 7	I to M: 9
J to A: 22	J to B: 24	J to C: 25	J to D: 30	J to E: 31	J to F: 28
J to G: 30	J to H: 18	J to I: 2	J to K: 8	J to L: 9	J to M: 11
K to A: 35	K to B: 37	K to C: 38	K to D: 43	K to E: 44	K to F: 41
K to G: 43	K to H: 31	K to I: 15	K to J: 17	K to L: 22	K to M: 24