SYSC 4101: Software Validation
Lab 4
Juanita Rodelo, 101141857

Exercise:
Assumptions:

- I am assuming that the system will still return a price of an item even if it is out of stock
- I am assuming that we are only considering the value characteristic of the itemType parameter (not including length of value, no negative numbers)
- I am assuming that we are only considering 7 digit value for itemCode (no less than 7 digits, not more than 7 digits, no negatives)
- I am assuming the range of each individual digit in the itemCode parameter is 0 through 9
- I am assuming the weight is an int, not a double

Parameters:	Constraint:
• int itemType:	
○ value of int: (5 possible inputs)	
▪ B1: 0	
▪ B2: 1	
▪ B3: 2	
▪ B4: 3	
▪ B5: Other	[error]
• int itemCode:	
○ Valid/invalid	
▪ B6: valid	
▪ B7: invalid	[error]
• int quantity:	
○ Value of int:	
▪ B8: Less than 0	[error]
▪ B9: 0	[if itemType == 1 or 3 this is valid else error]
▪ B10: less than or equal to 50	[itemType==0 or 2], [B13 – no weight]
▪ B11: More than 50	[error]
• int weight:	
○ value of int:	
▪ B12: less than 0	[error]
▪ B13: 0 kg	[if itemType == 0, 2, 3 this is valid else error]
▪ B14: Less than or equal to 50 kg	[if itemType == 1], [B9 – no quantity]ba
▪ B15: More than 50kg	[error]

Base Choice Criterion Frame:

B1, B6, B10, B14

This is not feasible since a constraint for B10 is that B13 is there, which states that the weight is 0, meaning there is no weight. This is because we can't have a product that has a valid quantity and has a weight. This goes against the rules of the system. The same goes for the other way around, there is a constraint on B14 that states that B9 needs to be there, which says that the quantity of a product is 0. This is because we can't have a valid weight of a product and a quantity since the weight tells us the quantity. This also goes against the rules of the system. The criteria provided is limited so we this base choice test frame is not feasible.

Pair-Wise Criterion:

B1, B6

- B10, 13
- This pair only requires one combination: B1, B6, B10, B13 to create a feasible test frame

B2, B6

- B9, B14
- This pair only requires one combination: B2, B6, B9, B14 to create a feasible test frame

B3, B6

- B10, B13
- This pair only requires one combination: B3, B6, B10, B13 to create a feasible test frame

B4, B6

- B9, B13
- This pair only requires one combination: B4, B6, B9, B13 to create a feasible test frame

B6, B9

- Can be combined with B2 or B4 from first set of characteristics
- Can only be combined with B14 from last set of characteristics
- This pair can only be combined to make the following test frames to create a feasible test frame:
    - B2, B6, B9, B14
    - B4, B6, B9, B14

B6, B10

- Can be combined with B1 or B3 from first set of characteristics
- Can only be combined with B13 from last set of characteristics
- This pair can only be combined to make the following test frames to create a feasible test frame:
    - B1, B6, B10, B13
    - B3, B6, B10, B13

B6, 13

- Can be combined with B1, B3, B4 from first set of characteristics
- Can only be combined with B10 from third set of characteristics

- This pair can only be combined to make the following test frames to create a feasible test frame:
    - B1, B6, B10, B13
    - B3, B6, B10, B13
    - B4, B6, B10, B13

B6, B14:

- Can only be combined with B2 from first set of characteristics
- Can only be combined with B9 from third set of characteristics
- This pair can only be combined with the following test frame to create a feasible test frame:
    - B2, B6, B9, B14

B5, '-'

- Since B5 is an error, whatever we combine it with will produce an error so there are 32 different combinations of sets possible (1* 2* 4* 4 = 32). Some will not be feasible.

B7, '-'

- Since B7 is an error, whatever we combine it with will produce an error so there are 80 different combinations of sets possible (5* 1* 4* 4 = 32). Some will not be feasible

B8, '-'

- Since B8 is an error, whatever we combine it with will produce an error so there are 40 different combinations of sets possible (5* 2* 1* 4 = 40). Some will not be feasible

B11, '-'

- Since B11 is an error, whatever we combine it with will produce an error so there are 40 different combinations of sets possible (5* 2* 1* 4 = 40). Some will not be feasible

B12, '-'

- Since B12 is an error, whatever we combine it with will produce an error so there are 40 different combinations of sets possible (5* 2* 4* 1 = 40). Some will not be feasible

B15, '-'

- Since B15 is an error, whatever we combine it with will produce an error so there are 40 different combinations of sets possible (5* 2* 4* 1 = 40). Some will not be feasible

Each-block criterion:
B1, B6, B10, B13
B2, B6, B9, **B14**
B3, B6, **B10**, B13
B4, B6, **B9**, **B13**

B5,'-', '-', '-'

- Since B5 is an error(because it is an invalid itemType), no matter what blocks we combine it with, it will always produce an error

'-', **B7**, '-', '-'

- Since B7 is an error(because it is an invalid itemCode), no matter what blocks we combine it with, it will always produce an error

'-', '-', **B8**, '-'

- Since B8 is an error(because we can't have a value of 0 for the quantity of an item), no matter what blocks we combine it with, it will always produce an error

'-', '-', '**B11'**, '-'

- Since B11 is an error(because we can't have more than 50 for the quantity of an item), no matter what blocks we combine it with, it will always produce an error

'-', '-', '-', **'B12'**

- Since B12 is an error(because we can't have less than 0kg for the weight of an item), no matter what blocks we combine it with, it will always produce an error

'-', '-', '-', **'B15'**

- Since B15 is an error(because we can't have more than 50kg for the weight of an item), no matter what blocks we combine it with, it will always produce an error