There is one exercise in this assessment. If the requirements are unclear then please ask for clarification.

1) Data Structures and Algorithms Exercise

Thank you for agreeing to take the Data Structures and Algorithms programming exercise. This test is intended to allow you to demonstrate your programming skills and your ability to design and implement a simple application. The subject of this exercise does not necessarily reflect the application areas that the role will cover but is intended to check that you have the necessary skills.

The aim of the task is to write an application that processes a file containing data simulating the market updates happening on an exchange.

You will provide us with your source code which will be expected to compile as a normal SBT Scala project. The code should not need to depend on anything other than the Scala standard library. Successful candidates will be working as part of a team where integration and team work are highly valued, so a cleanly packaged and easy-to-run solution is preferred.

It will be run on input data different to that given in the example. It should be efficient and perform well for various book depth values.

This test is about simulating a simplified model of an order book on a financial exchange:
http://en.wikipedia.org/wiki/Order_(exchange)

The input file will be a list of events representing changes to the order book in the order that they happen.

The data file will contain 1 update per line, each line containing the following data separated by spaces:

Position	Name	Type	
1	Instruction	U,D or N	U=Update, D=Delete, N=New
2	Side	B or A	B=Bid, A=Ask
3	Price Level Index	Integer	Price Level Index of change in range 1..book_depth
4	Price	Integer	Price in ticks
5	Quantity	Integer	Number of contracts at price level

Output
Your application should produce, after processing all of the input data, the most recent order book (ie based on reading all market updates) as one price level per line.
eg:
Bid Price[1], Bid Quantity[1], Ask Price[1], Ask Quantity[1] …
Bid Price[n], Bid Quantity[n], Ask Price[n], Ask Quantity[n]
Where n is the book depth and price is in $.

Command Line
This application will have to take the following command line arguments:

- Filename – the name of the file containing the input data.
- Tick Size – a floating point number giving the minimum $ price movements.
- Book Depth – an integer giving the number of price levels to keep track of.

Market Update Instruction Types

- N: Insert a new price level. Existing price levels with a greater or equal index are shifted up.
- D: Delete a price level. Existing price levels with a higher index will be shifted down.
- U: This will contain the new values for an existing price level. An update will not be given for a price level that hasn't already been provided using a New instruction.

Book Levels

- It is only necessary to keep track of 'Book Depth' price levels as given on the command line.
- It is not necessary to keep track of price levels beyond 'Book Depth'.
- Price levels within the range 1..book_depth that have not been provided should have values of zero.

Example
updates.txt
N B 1 5 30
N B 2 4 40
N A 1 6 10
N A 2 7 10
U A 2 7 20
U B 1 5 40

prog updates.txt 10.0 2

Output
50.0,40,60.0,10
40.0,40,70.0,20