First Derivatives plc

Kx Technology Training Courses

Q fundamentals

Exercises

Last Modified: 15 January 2009

Contents

CONTENTS 2

THE BASICS 3

ADVANCED QUESTIONS 3

DATA TYPES AND STRUCTURES 4

LISTS 5

LIST MANIPULATION 5
INDEXING AND MIXED TYPE LISTS 5

DICTIONARIES 6

DICTIONARY ARITHMETIC 6

ENUMERATION 7

FUNCTIONS 8

EXERCISE A 8
EXERCISE B 8
ADVANCED QUESTIONS 9

ADVERBS 10

SCAN/OVER 10
EACH 10

IPC 11

EXERCISE A 11
EXERCISE B 11
FLUSHING 11
MESSAGE HANDLERS 12

TABLE CREATION / MANIPULATION 13

SIMPLE QUERIES 14

EXERCISE A 14
EXERCISE B 14

TABLE ARITHMETIC 16

JOINS 17

EXERCISE A 17
EXERCISE B 17
EXERCISE C 18
EXERCISE D 18
PIVOTING 18

IMPORTING DATA 20

.Z LIBRARY 21

ODBC 22

The Basics

1) Start a new q session

2) Calculate 2+3

3) Calculate 2*10+3.

4) Create a new variable x equal to 56.

5) Define y as x divided by 28

6) Write a dyadic function, fxy, which calculates the square of x divided by the square of y.

7) In a script, define a as 4.56, b as 9 and c as -21. Then define a function, fabc, which calculates the average of a, b, c. Load this script into q and run the command “fabc[a;b;c]” to ensure it has been successful

8) Return a list of all functions defined in the current workspace

9) Quit the q session

Advanced Questions

1) Calculate 100-2*(4+(10+1/2)) using as few characters as possible.

2) Calculate (((100-2)*(4+10))+1)/2 using as few characters as possible

Data Types and Structures

1) Compare the datatype values of 3, 3f, 3 1 2, 3 1 2f, and (3f;1;2j)

2) Cast “12:30:00” to time/minute/second

3) Cast 123.23 to real

4) Extract the month, week and year from the date 2007.12.25

5) Get the current time/date/minute/second/hour

6) Cast the string “xyz” to a symbol and concatenate to the end of the simple list
alpha:`abc`def`ghi

7) Replace the second item of the list alpha with `jkl

8) Cast 1234 to a string

9) Get “o” from “hello”

10) Get the 1st, 3rd and 5th element of this list: x:("abc";10 5 0;(2;33;(4 5));"blah";1 3)

11) Get first and second element of each element of x by eliding an index

12) Replace the 3rd item of x with (4;5) using only indexing and assignment

13) Get (("ab";10 5 0);(10 5 0;"ab")) by indexing into the list x

14) Make a dictionary containing the following information:
`a: 0 1 2
`b: "some text"
`c: 3.12345

15) View values of entries in dictionary and look at shape, type

16) Assign a new value of 21.1 to `c and change the last item of `a to 3

17) We can represent a matrix using nested lists. Define m as the 3x3 matrix containing the values 1-9, going from left-to-right then top-to-bottom. That is:

```
1 4 7
2 5 8
3 6 9
```

18) Find the transpose of this matrix

19) Replace the 8 in the matrix with another 2

Lists

List Manipulation

1) Create a 10 element list of type long called a. Find out the type of this list (as a number)

2) Create a different list of longs called b.

3) Create c, which is a joined to b.

4) Create d, which is the first 2 elements of a joined to everything except the first element of b.

5) Calculate the number of elements in c which are greater than 5.

6) Extract the elements of c which are greater than 5.

7) Sum the elements of c which are greater than 5

8) Update the elements of c which are greater than 5 to be the null long. Use both the re-assignment syntax and the amend syntax to amend in place.

9) Create two lists, a and b. a should contain 5 and b should contain 10 random shorts between 0 and 20.

10) Compute the intersection of a and b

11) Compute the union

12) Compute the ascending distinct items

13) Compute find the indices of b which are in the intersection of a and b

14) Compute the sum of the elements of b which are in the intersection of a and b (remember that items could be duplicated). What do you notice about the result?

15) Compute the elements of b which are not in a

Indexing and mixed type lists

1) Create a list, e, containing two strings, "hello" and "world"

2) Create a list called f containing the following elements :
    a) The symbols `ab and `bc.
    b) The number 12
    c) The list e.

3) Extract the symbol `bc from f.

4) Extract the character string "hello" from f.

5) Extract the character "r" from f

6) Use amend syntax to capitalise the e within f. (do not amend e – create a new variable)

7) Use amend-at-depth to capitalise the first 3 elements of “hello” within f.

8) As above, but do it for the first 3 elements of both hello and world.

Dictionaries

1) Create a simple dictionary, g, to map the first 5 type names (symbols) outlined in the table on http://kx.com/q/d/q1.htm to their corresponding number type.

2) Retrieve from g the number types of short and int.

3) Add the correct value for type date.

4) Delete the value for type boolean.

5) Recreate g so that for each entry it both the number type and the character type.

6) Use amend syntax to increment the number type of int.

7) From dictionary g, use both the take and drop syntax to create a dictionary containing only int, long and short.

Dictionary Arithmetic

1) Create 2 dictionaries, d1:`a`b`c!1 2 3 and d2:`a`b`d!2 2 5. What are their types?

2) Create dictionary d3, which is dictionary d1 updated with dictionary d2

3) Extract the keys where the values of d1 and d2 are equivalent.

4) Extract the keys where the summation of the values is greater than 3

5) Use join each (,’) to concatenate d1 and d2. Observe the result?

Enumeration

1) Define v as the list `c`b`a`c`c`b`a`b`a`a`a`c. Set u as the list of distinct values within this list. Then set ev as the enumeration of v, within the domain of u.

2) Update ev so that every `b becomes `x. How would you do this with v? What are the benefits of doing this using enumerations?

3) Add a new value, `d, to the end of ev

Functions

Exercise A

Let x be defined as 1056 14 145 7261 5496 1201 -9643

1) Create the integer sequence 60-80, i.e. 60 61 62 … 79

2) What is the average of x? What is the pairwise moving average of x? What is the median of x?

3) What is the weighted average of x if we apply the weightings 38 47 10 55 60 81 20?

4) What is the deviation of x? Check this by writing your own deviation function, dev2.

   The deviation of x can be defined as the sqrt of the variance of x. The variance of x is the square of the average of x subtracted from the average of "x squared"

5) Find the difference between successive pairs of x (should start -1052 141…). Check this using the "prev" or "next" command.

6) Return the highest value of x. Return the lowest value of x.

7) Return the value of x largest in magnitude (ignore sign) and the value of x smallest in magnitude

8) Return the signs of x. Check (true or false) if each member of x is non-negative

9) What is the product of the first 10 integers (the factorial)? Write a function capable of calculating the factorial of the first n digits, where n is the input.

10) Sum all the members of x together

11) Obtain a running sum/product of x

12) Define y as the remainder when x is divided by 12. Define z as the number of times 12 divides x. Check 12z+y=x

13) Round x/z down to the nearest integer.

14) Replace the nulls in (0N 14 66 4 22 66 0N 72 26 0N) with
    a) The average of the non-null values, rounded up to the nearest integer
    b) The value before

15) Select the non-null elements of (0N 14 66 4 22 66 0N 72 26 0N)

16) Extract the distinct values from the list `a`b`a`b`c`a`b`a`b`c`b`c`a

17) Use .Q.A to get the upper case alphabet. Convert this into the lower case alphabet. Then cut into 4-letter lists

Exercise B

1) Create a function, f, which takes two parameters and adds them. Execute f for two numbers.

2) Create a function, f1, which is equivalent to the mathematical function

   res = - (y*((x+1)^2))/ (((x+1)*2) - 1)

   Try to avoid using brackets as much as possible

3) Create a function, f2, which is a projection of f1 with the first parameter bound to 10. Execute f2 for a number.

4) It should be possible to execute f2 on a list of numbers, e.g. f2 til 10. It can also be executed on a list of numbers 1 by 1, using each, e.g. f2 each til 10.

   a) Confirm that the output from these two operations is equivalent.
   b) Using the inbuilt timer (\t) and a do loop, investigate the differing performance in executing f2 using each and f2 directly on the list. Which is faster? What happens as the list gets longer?

5) Define the following function

```
j:{[a] {a+1}[a] }
```

   a) Execute j[2]. What is the problem? Fix the problem.
   b) Execute j[`a]. What is the problem?
   c) Call j with argument `a in an error trap and return 0 upon failure.
   d) Call j with argument `a and print out the error text, then return 0 upon failure
   e) Call j with argument `a. Upon the error, increment the global variable “err” log a warning to standard error

Advanced Questions

1) Get “o” from “hello” in as many different ways using different q functions

2) Compute an estimator for π (pi) using random pairs (x,y) in a square plane of length 1 and taking into account that the ratio of the area of a quarter-circle to the area of the square is π/4. Write your solution in a q script and load it into memory.

3) Given a list x: 5 8 7 9 12 return the greatest difference between x[n] and x[n+1]

4) For the string “kdb plus is fun”
   a) Get the indices of blank spaces
   b) Get the indices of non-blank spaces
   c) Replace the blank spaces in the sentence with “_” so it reads “kdb_plus_is_fun”
   d) Return a 4 item list of symbols: `kdb`plus`is`fun

5) Given the above list of symbols return the string “Kdb Plus Is Fun” – note the first letters in the list are lowercase but in the result are uppercase

Adverbs

Scan/Over

1) Rewrite the formula (x^2) - 4x - 8 = 0 in the form x=f(x). Define this as a monadic q function and use over (/) to solve the equation. Start with an initial value of -2

2) Use the syntax N f/ x to generate the Nth row of a binomial tree (1;1 1;1 2 1;1 3 3 1;...)

3) Given a vector of numbers and a "minimum step size", using scan, write a function which will return

   result: 0 0 0 0 0 5 5 5 5 5 10 10 10 10 10

   given the inputs

   x: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 and s: 5

   OR

   result: 1 1 1 1 1 4 4 4 4 0 0 0 0 3 6

   given the inputs

   x: 1 2 1 2 3 4 3 4 2 0 2 1 2 3 6 and s: 3

   i.e. the vector stays constant until the change is greater than the minimum step size

4) Assume that interest rates are compounded daily, and there are 365 days in a year. Create a dyadic function, f, which calculates the value of initial deposit x after accruing interest for one day at rate r.

5) Calculate the value of $100 after one day, given an interest rate (IR) of 5%.

6) Using / (over), calculate the value of $100 after 12 days at IR 5%.

7) Using \ (scan), generate the series of 12 days' compounding.

8) How many days will it take for the $100 to become $101?

9) Calculate the value of $100, $256 and $9872 after 20 days. Do this in one call.

10) The interest rate is no longer fixed. Calculate the times series of values of $100 over 5 days given interest rates of 5%, 5.2%, 5.8%, 6.5%, 5.9% on each day.

11) Calculate the value after 5 days of the initial deposits from question 6 using the interest rates in question 7.

Each

1) Join (1 2 3;"abcd") and (4 5;"ef") so that numbers are together and letters are together.

2) Now reverse each item of the list so that the integers descend and the alphabet runs backwards but the items stay in the same place within the list.

3) Put a comma after each letter of the list.

4) Divide 10 by each integer in the list.

5) Create a multiplication matrix. That is, a matrix whose (ij) entry is i*j.

IPC

Exercise A

1) Start one process on a port of your choice. Start another and open a connection to the server process.

2) Create a sample trade table on the server instance and insert a single row, sending the message as a string – just time, sym and price is fine.

3) Now insert multiple rows from the client using the second message format

4) Run a select query from the client and insert the results into a local variable.

5) From the client, create an empty quote table on the server – just a simple one with time, sym, bid & ask.

6) Define a simple function on the server and call it from the client.

Exercise B

1) Start a process which loads trade.q on port 5000

2) From another process, make a connection

3) Synchronously insert a row into the table, using string notation.

4) Synchronously insert a row, using functional notation

5) Synchronously send the following string “1+`a”. What happens?

6) Repeat (d) but do it asynchronously. What happens?

7) Put the server (port 5000) into error trap mode and repeat e. What happens?

8) Modify the server so it always returns 0 to any synchronous request.

Flushing

Using the script IPC/ExKill.q

Deleted:

```
x:.z.x;portnum:x 0;
processes:`int$();

processes,:"I"$portnum;

password":qprocess:inter"

kill:{
        h : @[hopen;`$"::",(string x),password;{-1"error : ",x; 0}];
        if[not h = 0;
                (neg h)(.:;"\\\\")]}



kill each processes

\\
```

Formatted: Indent: Left: 1.27 cm

1) Start up one process. In another window use the script to kill the initial process you started.

2) What happens?

3) How can this be fixed? I.e you are still sending the asych kill. Clue: Now in 2.4 it can be fixed both synchronously and asynchronously.

4) What would be a better way to do this? Answer: an asynchronous kill.

Message Handlers

1) Modify the message handlers so that when a user opens or closes a connection, the information is stored in a table. The table should have the following columns: handle, time_of_connection, username, IP, time_of_close.

   Hints:
   .z.pc takes the message handler as its parameter
   "."sv string "i"$0x00 vs .z.a gives a more readable version of .z.a (IP address)

2) Modify .z.pg and .z.ps to create some simple logging so that all calls are inserted into an operations table. The operations table should contain the following columns: username, operation, duration, synch_or_asynch.

3) Modify .z.pg and .z.ps again so that instead of storing the results in a table they store it in a pipe delimited text file

4) Make a further modification so that console input is also included in the operations table.

5) Add http logging so that all connection and operations coming through the web interface are also recorded.

6) Finally, change the authentication so that if a user attempts to open a connection to the session the password is checked. If the password is not alphanumeric (i.e. contains at least one letter and one number) do not allow the connection, instead return the error `alphanumeric.

Table Creation / Manipulation

Using the table t1:([] sym:`a`b`c`d; price:1 2 3 4f)

1) Use the insert syntax to insert the symbols `e->`g and the prices 5->7. Do it as a bulk insert.

2) Use the join (,) syntax to do the same thing

3) Extract the 3rd row from t1. What is its type?

4) Extract the distinct items from t1

5) Using the distinct items from t1 as defined above, create t2 which is t1 with an additional column, size. Size should be of type int.

6) Make sym be the key of t1 (xkey keyword)

7) Run the following code

```
t2 upsert flip (`a`g;1 2;3 5)
```

Why does it fail? Make it work.

8) Consider the two following tables

```
t3:([] sym:`a`b`c`d; price:1 2 3 4f;size:100 200 300 400)
t4:([] sym:`a`c`f; price:11 3 11f;size:100 300 1000)
```

Retrieve:

a) the rows where t3 and t4 are the same
b) the row indices of t3 which are in t4
c) the union set of t3 and t4
d) the rows which are in t3 but not t4

9) Key t3 and t4 by sym. Extract the symbols where the sizes match between the two tables.

Simple Queries

Exercise A

1) In a q session, create a table, trade, containing date, time, sym, side, price fields. Fill the table with 10000 random records with the following constraints:
   - 3 different dates (e.g. today, yesterday, day before yesterday)
   - random times (e.g. between 00:00:00.000 and now)
   - 5 different syms: `VOD.L`BMW.DE`AAA.L`FDP.L`GOOG.NY
   - 2 sides:`B`S
   - price between 0 and 200

2) Select from trade all buy side trades (`B) made by `AAA.L on one particular date.

3) Repeat 2 applying the where constraints in the order side, sym, date. How long does it take to do this 1000 times? Repeat in the order date, sym, side. How long does this take?

4) Select all trades made on a given date with a price between 100 and 110

5) Generate a count of the number of trades each sym made with the same conditions as 4

6) On the third day, `AAA.L was renamed to `BBB.L. Update trade to reflect this change

7) Delete all trades where the price was greater than 190

8) Get a list of syms, that traded on the last 2 days, that end in .L

9) By sym, calculate the maximum, minimum and average price as well as the spread (max-min) of trades made in the first minute of the second day on the buy side only.

Exercise B

1) Put the following code into a script

```
n:100000
t:([]time:asc n?0t; sym:n?`a`b`c`d`e;price:n?100e;size:n?1000)
show t
```

   Name the script with an appropriate .q extension and run it.

2) Select from the table the following :

3) All the times for the symbol `b;

4) All the data where time price is greater than 100

5) All the data where the price is greater than 100 and the size is less than 500.

6) The maximum price.

7) All the data where the time is within 12:00 and 13:00

8) The maximum price for each sym.

9) The maximum price for each sym and in each 5 minute time bucket (xbar keyword)

10) The open, close, high and low for each symbol and for each hour.

11) The vwap for each stock. (wavg keyword)

12) Add a column containing the running sum of the size.

13) Add a column containing the running sum of the size for each sym.

14) Add a column containing the running sum of the size for each sym in 10 minute time buckets.

15) Add a column containing the running vwap for each sym.

Table Arithmetic

Populate table trade as below:

```
trade:([]time:`time$();sym:`symbol$();price:`float$();size:`int$())
syms:`IBM`GS`FD`KX
n:1000
insert[`trade;(n?"t"$.z.Z;n?syms;10*n?100.0;10*n?100)]
```

1) Multiply each price by 2.

2) Add 10.00 to the trade price where the symbol is FD.

3) Find the total running sum of shares traded from a particular basket of shares.

4) Find the running sym of shares traded by sym from a particular basket of shares.

5) Find the average price a bucket of symbols is trading at throughout the day, not taking volume into consideration.

6) Find the average price a symbol is trading at throughout the day, not taking the volume into consideration

7) Find the 3 step moving average of the price by sym, not taking the volume traded into consideration.

8) Find the average price traded throughout the day from a trade table, taking the volume of each trade into consideration.

9) Find the VWAP per sym throughout the day from a trade table.

Joins

Exercise A

1) Create a table, trade with columns: sym,price,size, of count 10. Sym column data should be generated from the following syms `` `A`B`C`D `` and suitable values should be chosen for price and size. Define the following table:
   industry:([sym:`A`B`C`E];ind:`IT`Finance`Media`Transport)

2) Join the industry and trade tables to produce a table which is the intersection of the tables.

3) Join industry and trade so that an ind column is added to the trade table. All original entries that were in trade should still be present where there is no corresponding ind for a sym there should be a null value in the ind column. Call this table trade2.

4) Using lj join newInd:([sym:`A`C]ind:`Transport`Healthcare;Ex:`N`P`X) and trade2. Describe how the returned table differs from trade2.

5) Using ij join newInd and trade2. Describe how the returned table differs from trade2

6) Given:

```
tab0::([]a:1 2 3 2;b:`x`y`z`x)
tab1:([]a:3 3 2 2;c:`a`b`c`d)
```

   Produce:

```
a c b
-----
3 a z
3 b z
2 c y
2 c x
2 d y
2 d x
```

7) Below is table containing a record of changes in the telephone numbers of Tom and Bob during the last year.

```
phoneNum:([Name:`Tom`Bob`Tom`Bob;Date:2007.06.01 2007.06.01
2008.01.01 2008.06.01]phNum:336699 123456 999999 778899)
```

   Using phoneNum produce a table that shows Bob’s telephone number on 2007.06.02, 2008.01.01, 2008.01.02 and today.

Exercise B

1) Create a table, medals, with columns: year,color,country. For each year (1988,1992,1996,2000) there should be 3 colors: gold,silver,bronze. For each of these 12 records allocate a country from the choice `CHN`GER`KOR`SWE randomly.

2) Create a second table, country1, containing 2 columns id, name. The table should be keyed on id and should have a count of 5. Choose the ids `ALG`CHN`GER`KOR`SWE and names Algeria, China, Germany, Korea, Sweden

3) Using lj create a table containing the year, color of medal, and name of country that won that medal.

4) Using the table created in 3, or otherwise, calculate how many gold medals Korea won.

5) If a gold counts for 3 points, silver for 2 and bronze for 1, how many points did each country score over the 4 years?

Exercise C

1) Consider 3 train stations: Waterloo, Paddington, Euston. Each of these cover an area of the city: South, West, North respectively. Represent this information in a table, Trains, with columns Station and Direction.

   Now consider 3 train companies: Southwest, GNER, Virgin with the respective train colours Blue, Green, Red. Represent this information in a table, Management, with columns Company and TrainColour.

2) Create a single table, Transport, from these two tables where the first record in Trains gets matched to the first record in Management.

3) Make Transport a keyed table, using Station as the key.

4) Create a table, Journeys, with the fields Start, End, Price and 10 entries. Start should be populated with a random selection from Waterloo, Paddington and Euston. End and Price should be appropriately set

5) Using a suitable type of join, link Transport to Journeys so that we have Start, End, Price, Company

6) Southwest decide to start 2 new services from Euston to Glasgow and Bristol priced at 21.1 and 3.14 respectively. Make a table NewJourneys with the same schema as in 5 (i.e. Start, End, Price, Company) to reflect this. Then join this table to the table created in 5

Exercise D

Run the following code to generate a trade and quote table:

```
s:`IBM`WMT`GS`LEH; x:`O`N`Q
n:10000; nt:n; nq:4*n;
trade:([]time:asc nt?0t;sym:nt?s;price:50+0.1*nt?5000;size:nt?1000;ex:nt?x)
quote:([]time:asc
nq?0t;sym:nq?s;bid:50+0.1*nq?5000;ask:50+0.1*nq?5000;bsize:nq?1000;asize:nq?100
0;ex:nq?x)
```

1) Use an asof join to return the prevailing quote for every sym at the beginning of every hour

2) For each trade in the trade table, find the change in price for that symbol in the 5 minutes preceding the trade

3) For each trade in the trade table, find the vwap (for the symbol) for the 10 minutes following the trade

Pivoting

1) Consider the following table:

```
tab:([] c0:`a`b`c`a`b`c;c1:`b`a`a`c`c`b;c2:1 2 3 4 5 6) ;
```

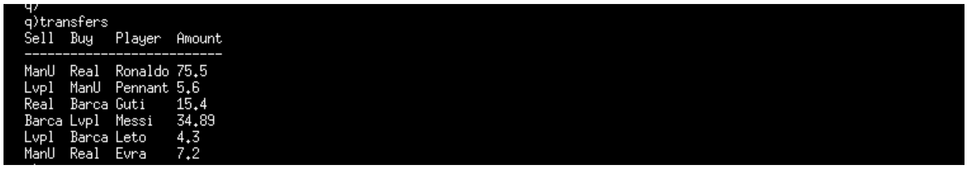
   So that c0 and c1 only have the values `a, `b or `c, and c2 is numeric. How can this pivot table be constructed?

```
c0\c1 a    b    c
```

```
-----------------------
a    0    1    4
b    2    0    5
c    3    6    0
```

i.e. c0 represents the rows and c1 the columns.

2) Now load in the script **transfers.q** and look at the table



```
q)transfers
Sell  Buy   Player  Amount
--------------------------
ManU  Real  Ronaldo 75.5
Lvpl  ManU  Pennant 5.6
Real  Barca Guti    15.4
Barca Lvpl  Messi   34.89
Lvpl  Barca Leto    4.3
ManU  Real  Evra    7.2
```

Using the construction from the previous example, how could we construct a pivot table where the data values were the transfer amounts from the above table?

3) How could we include both of the transfers from ManU to Real in Exercise B?

Importing Data

1) Import a text file. The file “trade.csv” is a comma-separated text file with the following fields:

   - date (list of dates)
   - sym (list of symbols)
   - size (list of integers)
   - price (list of floating-point values)
   - cond (list of characters)

   Import this file as a table into a q session, including all columns from the source file.

2) The folder “files” contains a series of csv files which contain trade data (the columns are the same as in exercise 1). Import each of the csv files and join the data into a single table. Sort the rows by date.

3) The file ‘test.mdb’ is a Microsoft Access file. Import all tables from this file into the q workspace.

4) Open an odbc connection to the file ‘test.mdb’. List all the tables in the file, then retrieve all rows from the table ‘Table1’ where the value of the ‘size’ column is 1000, then close the connection.

.z Library

1) Using q find out your computer's IP address. ("I"$0x0 vs … can be used to convert it to a more readable format)

2) Set a timer event to show the local time and gmt every minute.

3) Check your os, q version and release date

4) What PID does your q process have?

5) Make the necessary adjustments so that whenever a value is set a message is displayed. The message should be of the format:

   "At [time] on [date] the variable [variable](index [index]) was assigned the value [value]. This may also have an impact on [dependents]"

   Once this is working, see if you can instead write to a logfile so that the assignment can be replayed at a later date – such as in the case of a crash.

6) Modify your q session so that upon exiting it displays the message "goodbye" and appends the time to the file "logout.txt".

ODBC

1) Start a q process running on port 5001. Create a table, trade, with date, sym, price, side columns. Populate the table with 1000 records, with each field set appropriately

2) Using ODBC connect to the q process from excel and populate a worksheet with the data from trade.

3) Modify the query string so that only trades on the buy side are imported into excel.

4) In q, update the table to add a column cashflow that is equal to price if the side is buy, but is the negative of the price if the side is sell. Refresh the Excel data so it now contains this extra data

4) Using Macro Recorder if necessary, write a VBA macro that prompts the user for a sym and then imports all trades involving that sym (Tip: use InputBox to prompt the user)