TASK 4 - Parsing Messages (Part 1)

TASK 4 - Parsing Messages (Part 2)

TASK 3 - Feedhandler Concepts

NECESSARY KNOWLEDGE

- Creation and manipulation of dictionaries

I. AIM

The aim of this section is to take a message such as might be supplied by the simulator and parse it to return a dictionary containing the relevant information.

II. INSTRUCTIONS

Find the function p in fh.q it is also below:

```
p:{[s]
 l:"I=;"0: s
 };
```

To fully understand the operation of this function it may be necessary to read about the 0: function on the kx reference wiki.

Example:

```
q) g:p "1=`vod;2=23;4=12313.0"

q) 0N!g
```

Would return

```
(1 2 4;("`vod";"23";"12313.0"))

1      2    4
"`vod" "23" "12313.0"
```

In a nutshell the 0: function takes two parameters, that is to say it is dyadic. The first parameter is the parser definition as a three character string, "I=;" in this case. The first character is the type to cast FIDs to, integer "I" for this case. Possible values are S(ymbol) and I(nteger). The second character is the "*fid-from-value*" delimiter, the "=" in our example. The third is "*fid-value-pairs*" delimiter - the final ";".

The second parameter is the string to be parsed/converted. The return value is a nested list, with first list being the fids and second being the related values.

As the code above stands it will return two lists, one containing the FIDs and the other containing the associated values. However this function should return a dictionary with the FIDs mapped to values, modify the p function so it returns a dictionary containing the information.

III. TESTING

Launch the feedhandler with the command:

```
q fh.q
```

To test the p function enter the following from the q prompt:

```
p "1=`vod;2=23;4=12313.0"
```

The function should return

```
1| "`vod"
2| "23"
4| "12313.0"
```

Which sould be of type 99h.