COMP105 Lecture 20

Assignment 2

Assignment 2

Assignment 2 is **out now**

- The handout is available on the Canvas site

The deadline is

Thursday the 10th of December (week 9) at 12:00 midday

Assignment 2

The assignment is to use **higher-order programming**

- to process the transaction history of a stock portfolio

Some questions **require** you to use certain functions

- Q2 requires `map`
- Q4 requires `filter`
- Q7 requires `foldr`

Otherwise, there are **no restrictions** on what functions you can use

Transactions

A **transaction** is a tuple

```
('B', 100, 1104,  "VTI",  1),
```

- The first element is whether we `'B'`ought or `'S'`old
- The second element is the number of `units`
- The third element is the `price`
- The fourth element is the `stock`
- The fifth element is the `day`

Transactions

A **transaction log** is a list of transactions

```
type Transaction = (Char, Int, Int, String, Int)

test_log :: [Transaction]
test_log = [('B', 100, 1104,  "VTI",  1),
            ('B', 200,   36, "ONEQ",  3),
            ('B',  50, 1223,  "VTI",  5),
            ('S', 150, 1240,  "VTI",  9),
            ('B', 100,  229, "IWRD", 10),
            ('S', 200,   32, "ONEQ", 11),
            ('S', 100,  210, "IWRD", 12)
            ]
```

Part A (worth 30%)

Part A is to build a program to report the **trade history** on a given stock

- Using `map` and `filter`

```
ghci> trade_report "VTI" test_log
"Bought 100 units of VTI for 1104 pounds each on day 1\n
 Bought 50 units of VTI for 1223 pounds each on day 5\n
 Sold 150 units of VTI for 1240 pounds each on day 9\n"
```

Part B (worth 30%)

Part B is to build a program to calculate the **profit** for a given set of stocks

- Using `foldr`

```
ghci> profit_report ["VTI", "ONEQ"] test_log
"VTI: 14450\nONEQ: -800\n"
```

Computing profit

The profit is the amount of money that we make

- We **spend** `units * price` pounds when we buy
- We **gain** `units * price` pounds when we sell

So if we

- Bought 10 units at 100 pounds
- Sold 5 units at 110 pounds
- Sold 5 units at 120 pounds

Then our profit is

```
-(10 * 100) + (5 * 110) + (5 * 120) = 150
```

Part C (worth 40%)

Part C is to build a profit report program

- But the data is **given as a string**

```
BUY 100 VTI 1
BUY 200 ONEQ 3
BUY 50 VTI 5
SELL 150 VTI 9
BUY 100 IWRD 10
SELL 200 ONEQ 11
SELL 100 IWRD 12
```

Part C (worth 40%)

The prices are given in a **price database**

```
type Prices = [(String, [Int])]

test_prices :: Prices
test_prices = [
                ("VTI", [1689, 1785, 177, 1765, 1739, 1725,
                ("ONEQ", [201, 203, 199, 199, 193, 189, 189
                ("IWRD", [207, 211, 213, 221, 221, 222, 221
              ]
```

Part C (worth 40%)

Given a log string and a price database

- Compute a profit report like in part B
- Should include **all** stocks in the database

```
ghci> complex_profit_report test_str_log test_prices
"VTI: -7600\nONEQ: -2600\nIWRD: -500\n"
```

Penalties

There is a **5% penalty** for not following these guidelines:

- Your code should compile with no errors
- Do not alter the type signatures in the template file in any way
- Do not remove the module line
- Leave the stub error-messages if you do not answer a question

A **checker** is available for assignment 2