METU NCC,
CNG 242, Spring 2010,
Homework 3

Due 21st April 2010

In this homework you will implement an abstract data type in **Haskell** called `Bill`. `Bill` datatype will keep track of a customers bill. At each bill there are 0 or more orders (consider a restaurant). Customer can order 1 or more pieces of any item. Each item has a unique unit price. Customer can order as many items s/he wishes. Items are given arbitrary string names. All strings operations are case sensitive.

You are required to implement the following module in Bill.hs:

```
module Bill( Bill , newbill , get, add, quickadd, cancel, merge, billsum, show) where
```

Bill

It is the abstract data type for all bills. The details can be anything you've chosen, but not exported.

newbill :: Bill

It is a constructor returning an empty bill without any order.

add :: String → Int → Float → Bill → Bill

It adds an order to current bill and return as a new bill. The first parameter (String) is the name of the item, second (Int) is the number of pieces for the item, third (Float) is the unit price of the item. The forth is the current bill. If there exists an order with exact same name, the new order number of pieces is added to the current number of pieces and price is updated as the new price in third parameter. If no such order exists, it is inserted with is number of pieces and price values.

For example `t1 = add "pizza"2 3.4 newbill` adds 2 pieces of pizza with 3.4TL each on an empty bill and returns the bill.
`t2 = add "pizza"3 3.2 t1` adds 3 more pieces of pizza to previous value. The resulting value has 5 pieces of pizza and price of each piece is 3.2 (later overrides).
`t3 = add "pasta"1 3.2 t1` adds 1 piece of pasta in the first value. The resulting bill has 1 piece of pasta (3.2TL) and 2 pieces of pizza ($2 \times 3.4 = 6.8$TL) ordered.

get :: String → Bill → Maybe (Int, Float)

It gets the current state of order for the given name. First parameter is the name of the item and the second parameter is the bill. It returns `Nothing` if name does not exists as an order in the bill. It returns, `Just (count, unitprice)`, the number of pieces the name is ordered and the unit price in a tuple enclosed in `Just` if the name exists as an order.

`get "pizza"t2` returns `Just (5, 3.2)`
`get "pasta"t2` returns `Nothing`

quickadd :: String → Bill → Bill

It simply increments the number of pieces of the first parameter name. If name exists in the second parameter bill, the exact bill with the number of pieces for the name is incremented. If it does not, it generates an error. In order to produce the error, if name does not exist in the bill, evaluate '`error "not ordered yet"` . It will cause a error and stop executing the function.

cancel :: String → Int → Bill → Bill

It cancels some of the orders from the bill. The first parameter is the name of the item and the second parameter is the number of items to be cancelled. It returns the new bill with orders cancelled. If the name has larger number of pieces already ordered, the number of pieces are decremented by the second parameter. Otherwise, if order is less than equal to the parameter, all orders of given name are deleted. There is no error checking about if the parameter is larger than the existing number of pieces.

`cancel "pizza"1 t3` returns a bill with 1 piece of pizza and 1 piece of pasta.
`cancel "pizza"2 t3` returns a bill with 1 piece of pasta.
`cancel "pizza"3 t3` returns a bill with 1 piece of pasta.
`cancel "pasta"200 t1` returns just `t1`.

merge :: Bill → Bill → Bill

It merges two bills into a bill that contain both orders. The unit prices will be taken from the second parameter bill in case of conflict.

`merge t2 t3` returns a value with 1 piece of pasta and 7 pieces of pizza (3.4TL each, from `t3`).

billsum :: Bill → Float

It calculates and returns the total price of the orders. It is sum of all item prices and each item price is the number of pieces multiplied by the unit price.

`billsum t3` returns 10.0 ($1 \times 3.2 + 2 \times 3.4$)

show :: Bill → String

Since abstract data type details are hidden, you cannot show it as usual with tags and values. This function returns a string to display content of the bill. You need to implement as:

```
instance Show Bill where
        show ... =
```

This will cause your values to be displayed on terminal in this way you want. You can use `'\n'` to set end of line. The following is the format to use for your bills:

```
Name                    Pieces  U.Prc.  Price
-----------------------------------------------
ayran                        3    1.40   4.20
coffee                       2    2.00   4.00
orange juice                 1    4.00   4.00
pasta                        2    3.00   6.00
pizza                        2    3.40   6.80
puding                       1    3.10   3.10
-----------------------------------------------
Price:                                   23.81
VAT:                                      4.29
Total:                                   28.10
```

In order to get formatted string and numerical values:
`import Text.Printf`
and use `printf "%8.2f" unitprice` to convert floating point value to formatted string. See the following example, a multiplication table implementation to have an idea.

```
module MTable(MTable, mtable, show) where
import Text.Printf
data MTable = M Int
mtable n = if n < 1 then error "positive value required"
           else M n

instance Show MTable where
  show (M n) = "m \\ n| " ++ showcols 1 n ++ "\n" ++  -- column headers
               take (n*7+4) (cycle "-")  ++     -- line from dashes
               showlines n  where
    -- "showlines m" draws lines 1 to m
    showlines 0 = ""
    showlines m = (showlines (m-1)) ++ "\n" ++ printf "%4d | " m ++ (showcols m n)

    -- "showcols r c" draws columns 1 to c of row r
    showcols row 1 = printf "%4d" row     -- this is the first column, the row number
    showcols row col = (showcols row (col-1))  ++ " | " ++  (printf "%4d" (row*col))
```

Please note that the items are sorted based on names during output. You can either keep them sorted or sort during `show`. Show also displays the total sum, VAT (Value added tax, which is $\frac{sum*0.18}{1.18}$) and total sum without VAT. Orders are kept in menu prices, including VAT, then VAT calculated and shown during `show`. For this calculations you need to multiply `Int` to `Float`. You can use `fromIntegral` to convert to float like:
`(fromIntegral (3::Int)) * (3.2::Float)`

Sample code:

```
module HW2 where
import Bill

t1 = add "pizza" 2 3.4 newbill
t2 = add "pasta" 2 3.0 t1
t3 = add "orange juice" 1 4.0 t2
t4 = add "ayran" 3 1.4 t3
t5 = add "puding" 1 3.1 t4
t6 = add "coffee" 2 2.0 t5

u1 = add "soup" 2 1.4 newbill
u2 = add "pide" 1 2.5 u1
u3 = add "orange juice" 1 3.5 u2
u4 = add "ayran" 1 1.0 u3
u5 = add "adana kebap" 1 3.0 u4
u6 = add "coffee" 2 1.5 u5

a1 =add  "orange juice" 2 3.5 u6
a2 =add  "tea" 2 1.0 u6
a3 =add  "orange juice" 2 3.5 t6
a4 =add  "tea" 2 1.0 t6

c1 = cancel  "tea" 2  t6
c2 = cancel  "puding" 2  t6
c3 = cancel  "ayran" 2  t6

g1 = get "orange juice" t6
g2 = get "apple juice" t6

q1 = quickadd "ayran" t6
q2 = quickadd "ayran" (quickadd "ayran" t6)
q3 = quickadd "soda" t6

b1 = billsum t6
b2 = billsum u6

m1 = merge t6 u6
m2 = merge u6 t6
```

Post your submission on METU Online. You can ask questions on METU Online forums. Note that all homeworks should be result of you individual work. No code sharing is allowed. Cheating policy: cheaters (both parties) take 0 from all 5-6 homeworks. Commissioning/trying to commission someone else to do your homework is subject to disciplinary action.

Sample Run:

```
*HW2> :browse Bill
data Bill
newbill :: Bill
get :: String -> Bill -> Maybe (Integer, Float)
quickadd :: String -> Bill -> Bill
add :: String -> Integer -> Float -> Bill -> Bill
cancel :: String -> Integer -> Bill -> Bill
merge :: Bill -> Bill -> Bill
billsum :: Bill -> Float
class Show a where
  ...
  show :: a -> String
  ...


*HW2> u6
Name                    Pieces  U.Prc.  Price
-----------------------------------------------
```

```
adana kebap                 1     3.00     3.00
ayran                       1     1.00     1.00
coffee                      2     1.50     3.00
orange juice                1     3.50     3.50
pide                        1     2.50     2.50
soup                        2     1.40     2.80
-------------------------------------------------
Price:                                     13.39
VAT:                                        2.41
Total:                                     15.80

*HW2> t6
Name                      Pieces  U.Prc.   Price
-------------------------------------------------
ayran                       3     1.40     4.20
coffee                      2     2.00     4.00
orange juice                1     4.00     4.00
pasta                       2     3.00     6.00
pizza                       2     3.40     6.80
puding                      1     3.10     3.10
-------------------------------------------------
Price:                                     23.81
VAT:                                        4.29
Total:                                     28.10

*HW2> add  "orange juice" 2 3.5 u6
Name                      Pieces  U.Prc.   Price
-------------------------------------------------
adana kebap                 1     3.00     3.00
ayran                       1     1.00     1.00
coffee                      2     1.50     3.00
orange juice                3     3.50    10.50
pide                        1     2.50     2.50
soup                        2     1.40     2.80
-------------------------------------------------
Price:                                     19.32
VAT:                                        3.48
Total:                                     22.80

*HW2> add  "tea" 2 1.0 u6
Name                      Pieces  U.Prc.   Price
-------------------------------------------------
adana kebap                 1     3.00     3.00
ayran                       1     1.00     1.00
coffee                      2     1.50     3.00
orange juice                1     3.50     3.50
pide                        1     2.50     2.50
soup                        2     1.40     2.80
tea                         2     1.00     2.00
-------------------------------------------------
Price:                                     15.08
VAT:                                        2.72
Total:                                     17.80

*HW2> add  "orange juice" 2 3.5 t6
Name                      Pieces  U.Prc.   Price
-------------------------------------------------
ayran                       3     1.40     4.20
coffee                      2     2.00     4.00
orange juice                3     3.50    10.50
pasta                       2     3.00     6.00
pizza                       2     3.40     6.80
puding                      1     3.10     3.10
-------------------------------------------------
```

```
Price:                                29.32
VAT:                                   5.28
Total:                                34.60

*HW2> add  "tea" 2 1.0 t6
Name                    Pieces  U.Prc.  Price
-------------------------------------------------
ayran                        3    1.40    4.20
coffee                       2    2.00    4.00
orange juice                 1    4.00    4.00
pasta                        2    3.00    6.00
pizza                        2    3.40    6.80
puding                       1    3.10    3.10
tea                          2    1.00    2.00
-------------------------------------------------
Price:                                25.51
VAT:                                   4.59
Total:                                30.10

*HW2> cancel  "tea" 2  t6
Name                    Pieces  U.Prc.  Price
-------------------------------------------------
ayran                        3    1.40    4.20
coffee                       2    2.00    4.00
orange juice                 1    4.00    4.00
pasta                        2    3.00    6.00
pizza                        2    3.40    6.80
puding                       1    3.10    3.10
-------------------------------------------------
Price:                                23.81
VAT:                                   4.29
Total:                                28.10

*HW2> cancel  "puding" 2  t6
Name                    Pieces  U.Prc.  Price
-------------------------------------------------
ayran                        3    1.40    4.20
coffee                       2    2.00    4.00
orange juice                 1    4.00    4.00
pasta                        2    3.00    6.00
pizza                        2    3.40    6.80
-------------------------------------------------
Price:                                21.19
VAT:                                   3.81
Total:                                25.00

*HW2> cancel  "ayran" 2  t6
Name                    Pieces  U.Prc.  Price
-------------------------------------------------
ayran                        1    1.40    1.40
coffee                       2    2.00    4.00
orange juice                 1    4.00    4.00
pasta                        2    3.00    6.00
pizza                        2    3.40    6.80
puding                       1    3.10    3.10
-------------------------------------------------
Price:                                21.44
VAT:                                   3.86
Total:                                25.30

*HW2> get "orange juice" t6
Just (1,4.0)

*HW2> get "apple juice" t6
```

```
Nothing

*HW2> quickadd "ayran" t6
Name                     Pieces  U.Prc.   Price
-----------------------------------------------
ayran                         4    1.40    5.60
coffee                        2    2.00    4.00
orange juice                  1    4.00    4.00
pasta                         2    3.00    6.00
pizza                         2    3.40    6.80
puding                        1    3.10    3.10
-----------------------------------------------
Price:                                     25.00
VAT:                                        4.50
Total:                                     29.50

*HW2> quickadd "ayran" (quickadd "ayran" t6)
Name                     Pieces  U.Prc.   Price
-----------------------------------------------
ayran                         5    1.40    7.00
coffee                        2    2.00    4.00
orange juice                  1    4.00    4.00
pasta                         2    3.00    6.00
pizza                         2    3.40    6.80
puding                        1    3.10    3.10
-----------------------------------------------
Price:                                     26.19
VAT:                                        4.71
Total:                                     30.90

*HW2> quickadd "soda" t6
*** Exception: not ordered yet

*HW2> billsum t6
28.099998

*HW2> billsum u6
15.8

*HW2> merge t6 u6
Name                     Pieces  U.Prc.   Price
-----------------------------------------------
adana kebap                   1    3.00    3.00
ayran                         4    1.00    4.00
coffee                        4    1.50    6.00
orange juice                  2    3.50    7.00
pasta                         2    3.00    6.00
pide                          1    2.50    2.50
pizza                         2    3.40    6.80
puding                        1    3.10    3.10
soup                          2    1.40    2.80
-----------------------------------------------
Price:                                     34.92
VAT:                                        6.28
Total:                                     41.20

*HW2> merge u6 t6
Name                     Pieces  U.Prc.   Price
-----------------------------------------------
adana kebap                   1    3.00    3.00
ayran                         4    1.40    5.60
coffee                        4    2.00    8.00
orange juice                  2    4.00    8.00
pasta                         2    3.00    6.00
```

pide	1	2.50	2.50
pizza	2	3.40	6.80
puding	1	3.10	3.10
soup	2	1.40	2.80
Price:			38.81
VAT:			6.99
Total:			45.80