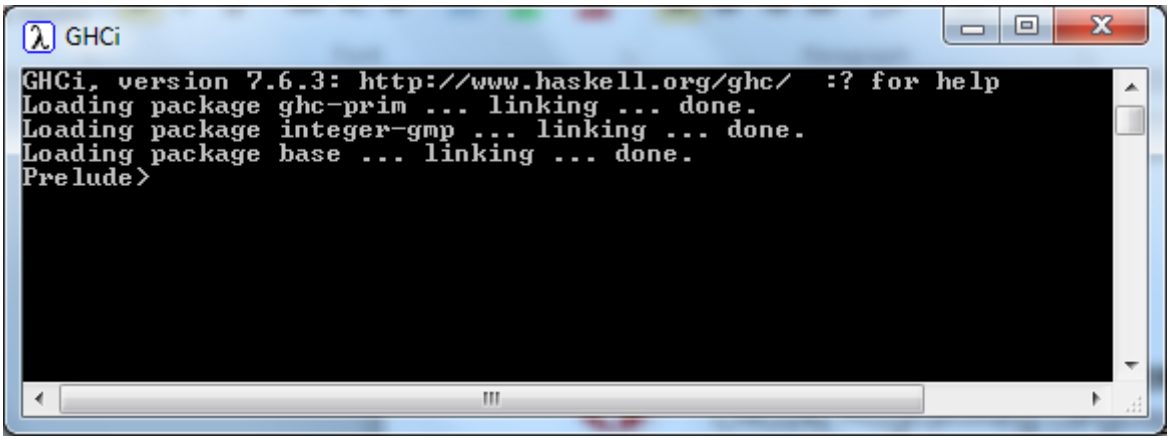
MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG242 Programming Language Concepts – Lab 1: Haskell

1. Starting Out

You can download the Haskell Platform from <http://www.haskell.org/platform/>
Start Button -> Programs -> Haskell Platform xxxx.x.x.x -> GHCi



2. Simple Arithmetic

```
Prelude> 3 + (5* (-2)) / 2
-2.0
Prelude> 5 / 2
2.5
Prelude> 3^4
81
Prelude> 16**0.5
4.0
Prelude> 16^0.5
Error!
```

3. Boolean Algebra

```
Prelude> True || False
True
Prelude> True && False
False
Prelude> True == False
False
Prelude> True /= False
True
Prelude> not False
True
```

4. Basic Built-in Functions

```
Prelude> min 4 5
4
Prelude> max 10 20
20
Prelude> min 7 (min 5 6)
5
Prelude> sqrt 16
4.0
Prelude> div 5 2
2
Prelude> 5 `div` 2
2
```

5. Simple User-defined Functions in the Command Prompt

In Haskell

```
Prelude> let myValue = 5
Prelude> myValue
5
```

Function Definition

Function Call

In C

```
int myValue () {
    return 5;
}

int main(void){
    printf ("%d", myValue());
    return 0;
}
```

Function Definition

Function Call

6. Simple User-defined Functions in a Haskell File

Some helpful commands:

```
:help || :?
:cd <dir>
:edit <file> || :e <file>
:load <file> || :l <file>
:reload || :r
```

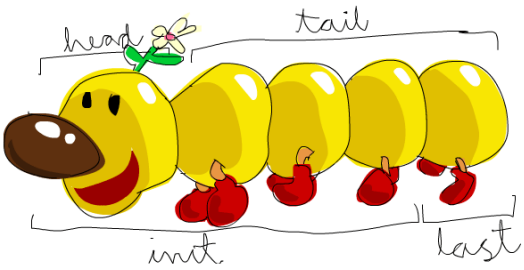
```
-- firstHaskell.hs

checkNumber x = if x >= 10
                then "It is equal or greater than 10"
                else
                "It is less than 10"

{- This is our first Haskell file
   which defines the checkNumber function
-}
```

Mandatory!

7. Lists

[1, 2, 3, 4, 5, 6, 7, 8]

```
Prelude> [1, 2, 3, 4] ++ [5]
[1,2,3,4,5]
Prelude> 0:[1, 2, 3, 4]
[0,1,2,3,4]
Prelude> ['H','e','l','l','o'] ++ (' ':['W','o','r','l','d'])
"Hello World"
```



[1]

[1] http://learnyouahaskell.com/starting-out#ready-set-go

```
Prelude> let myList = [1, 2, 3, 4, 5, 6, 7, 8]
Prelude> head myList
1
Prelude> tail myList
[2,3,4,5,6,7,8]
Prelude> init myList
[1,2,3,4,5,6,7]
Prelude> last myList
8
```

Some other helpful built-in functions for lists:

```
Prelude> length myList
8
Prelude> reverse myList
[8,7,6,5,4,3,2,1]
Prelude> minimum myList
1
Prelude> maximum myList
8
Prelude> sum myList
36
Prelude> product myList
40320
Prelude> take 2 myList
[1,2]
Prelude> drop 3 myList
[4,5,6,7,8]
Prelude> 4 `elem` myList
True
Prelude> 9 `elem` myList
False
Prelude> myList !! 2
3
```

8. Tuples

("CNG", 242)

```
Prelude> let myTuple = ("CNG", 242)
Prelude> fst myTuple
"CNG"
Prelude> snd myTuple
242
```

`fst` takes a pair and returns its first item.
`snd` takes a pair and returns its second item.

Practical Exercises

a. Write a Haskell function that gets a list which consist of numbers and returns the sum of the numbers in the list without the first and the last items.

b. Write a Haskell function that takes an item and a list. It then checks if the item exists in the list. If the item exists in the list, the function returns the list directly. If the item does not exist, the function adds the item to the list and returns the updated list.

c. Write a Haskell function that takes a tuple with three items and returns the third item.

d. Write a Haskell function that takes two lists and return the maximum value in these lists.

e. Write a function takes two points (x_1, y_1) and (x_2, y_2) with their x and y coordinates and calculate the distance between the points using the following formula. Hint: you can use sqrt function.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

f. [2]Anonymous Inc. has classified its employees into four categories, and has the following salary policy:

Class 1:	$10 per hour for regular hours and no overtime
Class 2 or 3:	$7 per hour for regular hours, and overtime hours at the rate of 1.5 times the rate for the regular hours
Class 4:	$5 per hour for regular hours, and overtime hours at the rate of 2.0 times the rate for the regular hours.

Write a Haskell function that takes an employee's classification and regular and overtime hours and returns the employee's pay. An error code (-1) should be returned if a classification number other than 1, 2, 3, or 4.

References

Miran Lipovača, *Learn You a Haskell for Great Good! A beginner's guide to Haskell*, No Starch Press, Daly City, California, United States, 2011

Useful Links

Learn You a Haskell <http://learnyouahaskell.com/chapters>

[2] Ram Kumar and Rakesh Agrawal, *Programming in ANSI C*, West Publishing Company, 1992 (This question is simplified and used as a Haskell question here)