1. Basic Built-in Functions

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

2. 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

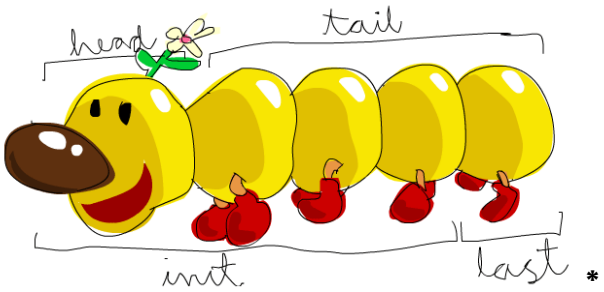
3. Simple User-defined Functions in a Haskell File ***

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"
```

4. Lists

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



* <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> elem 4 myList
True
Prelude> elem 9 myList
False
Prelude> 4 `elem` myList
True
Prelude> 9 `elem` myList
False
```

5. Tuples

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

References:

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

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 called “body” that gets a list which consist of numbers and returns the list without the first and last elements.

c. Write a Haskell function that takes two lists and returns the maximum value in these lists. For example, using following inputs: [3,2,5] [4,1,7] it should return 7

d. Modify the function above (c) to return a tuple that contains maximum values in these lists. For example, using following inputs: [3,2,5] [4,1,7] it should return (5,7)

e. 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.

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

g. Write a function takes two points as a tuple such as; (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.

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

h. 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 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.