

## Worksheet 1: *Basic Hugs*

---

Template file:	Worksheet1.hs
Labs :	29/1/2021
Topics:	Type synonyms. Pairs. Functions via patterns + Lists

---

Make sure that your submission compiles in GHCi.

Try to give a nice layout to your answers.

1. The daily costs incurred by a car plant is a function `cost` that depends on the number `n` of cars made per day. If  $0 \leq n$  and  $n \leq 500$  then the cost is  $5 * n + 1000$ . Otherwise the cost is  $10 * n + 450$ . Write a Haskell function coding this `cost` function.
2. 1 British Pound Sterling equals currently about 1.14 Euro. Write a function that converts a price in pounds into euros. And also a function which does the converse.  
  
(Euro sign can be typed using “Alt Gr” key (on the right of your space bar, in my case) together with 4)
3. An phone book for storing names and telephone numbers can be implemented in Haskell as follows.

```
type Name = String
type PhoneNumber = Int
type Person = (Name, PhoneNumber)
type PhoneBook = [Person]
```

- (a) (5 marks) Write the function `add :: Person -> PhoneBook -> PhoneBook` that adds an entry to the phone book at the beginning of the list.
- (b) (5 marks) Write the function `delete :: Name -> PhoneBook -> PhoneBook` that given the name of a person deletes all entries in the phone book with that name.
- (c) (5 marks) Write the function `find :: Name -> PhoneBook -> [PhoneNumber]` that gives the list of all telephone numbers of a certain person.
- (d) (5 marks) Write the function `update :: Name -> PhoneNumber -> PhoneNumber -> PhoneBook -> PhoneBook` that given the name and old phone number of a person updates that entry in the phone book with that the new phone number. (You may assume that the phone-book does not contain multiple entries of the same data.)

4. Read the section on Escaping text on <http://book.realworldhaskell.org/read/characters-strings-and-escaping-rules.html>.

Now write down a string `rawtext` such that `putStr rawtext` outputs exactly

```
"This is a \ \long string,  
\ \ spanning multiple lines,  
in fact 3 lines!"
```