

Functional Languages 11th practice

1. Create a new data type `Day` for days of a week.
2. Define a function `isFirstDayOfWeek` which returns `True` only for Monday.
3. Define a function `isWeekend` which returns `True` only for weekends.
4. Define a function `next` which returns the next day.
5. Define a data type `Time` which stores hour and minute. Use type synonyms where applicable.
6. Define a function for converting a `Time` to a `String`.

```
showTime (T 12 15) == "12.15"
showTime (T 12 05) == "12.5"
showTime (T 9 30)  == "9.30"
```
7. Define equality test between `Time` values.

```
eqTime (T 10 30) (T 10 30)
not (eqTime (T 10 30) (T 10 40))
not (eqTime (T 10 30) (T 11 30))
not (eqTime (T 10 30) (T 9 30))
```
8. Check whether a `Time` is strictly earlier than another.

```
isEarlier (T 10 15) (T 10 30)
isEarlier (T 10 35) (T 12 30)
not (isEarlier (T 12 30) (T 12 30))
not (isEarlier (T 10 35) (T 10 30))
not (isEarlier (T 13 30) (T 12 30))
```
9. Check whether a `Time` is between two others. `isBetween t1 t2 t3` if `t2` falls between `t1` and `t3` (either `t1 < t2 < t3` or `t3 < t2 < t1`).

```
isBetween (T 12 30) (T 12 40) (T 12 50)
isBetween (T 12 50) (T 12 40) (T 12 30)
not (isBetween (T 12 30) (T 12 15) (T 12 50))
not (isBetween (T 12 30) (T 10 00) (T 12 50))
not (isBetween (T 12 30) (T 15 00) (T 12 50))
```
9. Define a so-called smart constructor for `Time` which signals error on invalid hours and minutes.

```
time 12 60 --> "time: invalid minute: 60"
time 24 15 --> "time: invalid hour: 24"
time (-5) 2 --> "time: invalid hour: -5"
time 15 (-10) --> "time: invalid minute: -10"
eqTime (time 12 30) (T 12 30)
eqTime (time 23 59) (T 23 59)
eqTime (time 00 00) (T 00 00)
not (eqTime (time 12 30) (T 20 00))
```
10. Define a version of `Time` which uses 12-hour am-pm format. Let's call it `UStime`. Ask for equality test and conversion to `String` (make it instance of `Eq` and `Show`).
11. Define a function which converts `UStime` to `String`.

```
showUStime (AM 10 15) == "10.15 am"
showUStime (PM 2 30)  == "2.30 pm"
```
12. Convert `UStime` to `Time`. Recall that 12.00 am is midnight (00.00) and 12.00 pm is noon (12 in 12-hour time corresponds to 0 in 24-hour time).

```
eqTime (usTimeToTime (AM 10 15)) (T 10 15)
eqTime (usTimeToTime (AM 12 20)) (T 00 20)
eqTime (usTimeToTime (PM 2 30)) (T 14 30)
eqTime (usTimeToTime (PM 12 45)) (T 12 45)
```
13. Convert `Time` back to `UStime`.

```
timeToUStime (T 10 15) == AM 10 15
timeToUStime (T 00 20) == AM 12 20
timeToUStime (T 12 45) == PM 12 45
timeToUStime (T 15 10) == PM 3 10
```