Functional Languages 1st practice

1. Download the compiler from here and install.
2. Create a new file with `.hs` extension. Define an integer-valued variable.
3. Load the file in ghci.

   Hint: the load statement is `:l`. ghci also takes the file to be loaded as argument: `ghci First.hs`.
4. Define a string-valued variable.
5. Reload the file.

   Hint: the reload statement is `:r`.
6. Define a function which increases its integer parameter by one.

   ```
   inc 5    == 6
   inc 0    == 1
   inc (-5) == (-4)
   ```
7. Define a function which checks whether its parameter is even. Function `even` already exists in Haskell, so lets call our function `even'`.

   ```
   even' 2
   not (even' 3)
   even' (-4)
   ```

   Hint: the modulus function is `mod`.
8. Define a function which chcks whether its parameter is odd. Function `odd` already exists in Haskell, so lets call our function `odd'`.

   ```
   not (odd' 2)
   odd' 3
   not (odd' (-4))
   ```
9. Can you define `odd'` otherwise? Give a definition which calls `even'`.
10. Define a function which checks whether an integer divides another.

    ```
    2 `divides` 4
    not (4 `divides` 2)
    3 `divides` 9
    ```
11. Define a function which calculates the area of a rectangle using two sides.

    ```
    area 6 10 == 60
    ```
12. Check whether a triangle with three given sides can be drawn.

    ```
    triangleSides 2 1 2
    not (triangleSides 3 4 1)
    ```

    Hint: logic operators are `&&` and `||`. Relational operators are `> < >= <= ==` and `/=`.
13. Check whether three integers are Pythagorean triples.

    ```
    pythagoreanTriple 3 4 5
    pythagoreanTriple 5 3 4
    not (pythagoreanTriple 2 3 4)
    ```

    Hint: the power operator is `^` as in `2 ^ 3`.
14. Check whether a year is a leap year. A year is considered a leap year if it can be divided by 4 but not 100. However, years divisible by 400 are also leap years:

    - 1992, 1996, 2012, 2016 are leap years, as they are divisible by 4 but not 100.
    - 1700, 1800, 1900 are not leaps, as they are divisible by 4 and also 100.
    - 1600, 2000 are leap years, as they are divisible 100 but also 400.

    ```
    isLeapYear 1992
    isLeapYear 1996
    isLeapYear 1600
    isLeapYear 2000
    not (isLeapYear 1700)
    not (isLeapYear 1800)
    ```