Record types are a way of creating a type where names can be assigned to the data contained within the type. Recall the definition of a Vector:

```haskell
data Vector = Vector { x::Float, y::Float, z::Float }
```

We can then reference the fields (we get accessor methods) via the fieldnames (e.g. `x`, `y`, and `z`).

IO is a tool, built into Haskell to allow interaction with the user and the “outside” world.

For this assignment you will use a record data type and IO (in particular the `getLine` and `putStrLn` functions) to create a Book collection.

1) Define a data type, using record syntax, to define a Book data type with fields “name”, “author” and “publication year”.

2) Over ride the default show method to display a Book as

```
NAME written by AUTHOR in publication year
```

3) Write a method to construct a book called `mkBook`. It will have the following type

```
mkBook:: String → String → String → Book
```

4) Create an interactive program called `collectBooks`. You may have to write helper functions to assist in this. The `collectBooks` function will take an integer which indicates the number of books you will insert. The books should be created, one by one, by prompting the user (with details about what to enter) for each field in the Book type. Once collected a book should be stored in a list. The list of books, once the number of books specified has been collected, should be displayed using show.

Hand in a Haskell file `HW10_<YOURNAME>.hs`.

Be sure to test your methods.