This week you will write a program that calls two function—which you will also write yourself. This is to emphasize that we want to design complex programs in a modular fashion in order to keep individual modules small and thus easier to test and maintain.

In this exercise we will simulate a small online bookstore. You will write a program that lets the user search for one of several book titles, check whether the book is available in the online store, and if so, add it to the shopping cart. The user will also be able to display the content of the shopping cart on the screen.

1. Load the bookstore data into your program

Download the file data.py from the Lab Exercises page of the course website. This file contains the two lists `bkList` and `avList`. The first list contains a number of book titles and the second list contains "available" if the corresponding book is in stock with the bookseller and "unavailable" if it is not. Load the book data into your program file using `import`:

```
import data
```

You can now access the two lists as `data.bkList` and `data.avList`.

2. Write a search function

Create a file search.py and in it a function with the following signature.

```
searchItem(booklist,availlist,book)
```

This function should take a book list, a corresponding availability list (both loaded from the data.py file), and a book title `book` that the user would like to search for as the inputs. The return value of the function should be `True` if the book is available, and `False` otherwise.

The keywords `True` and `False` are used in python to indicate the truth value of a statement. In fact, you have used such truth values before implicitly, for example in `if` statements. For example, in the statement

```
if i==0:
    <more code>
```

the expression `i==0` takes the value `True` if i is indeed zero, and the value `False` otherwise. Just as the statement `i==0` takes a truth value in this example, your function `searchItem` will take a truth value as well.

In order to do a search for the book `book`, your search function will have to traverse the book list item by item and find the position at which the title of the required book appears. The availability list will contain the availability status of this book at the same list index.

Test your search function using the lists from data.py before moving on to the next task.

3. Write a display function

Create a file display.py and in it a function with the following signature.

```
printCart(cart)
```

This function should take as its only input a list named `cart`, where each item in the list is a string (a book title, for example). The function should print each string in the list in order and on a new line.

Test your shopping cart display function using the lists from data.py before moving on to the next task.

4. Write the main program

Create a new file and write a program that prompts the user to enter three book titles to add to their shopping cart. For each book title that the user enters, the program should call your search function `searchItem` to find out if the book is available at the online book seller and if so, add it to a list named `cart`. At the end, your program should print the contents of the cart using the `printCart` function written earlier. Remember to import all files that contain the data and the two functions you wrote before.

In order to prompt the user to input a book title, you will need to use the function `raw_input()`. This function will let the user type in a string and return it to be used in your program. An examples use of this function would be as follows.

```
s = raw_input()
```