Extra Credit 3 - MARC & Pymarc

This extra credit assignment will be very similar to Assignment 4. But instead of reading a `csv` file and producing a `json` file, you will read a `mrc` file and write to a `csv` file.

You should be able to run your file by calling the following:

```
python3 extra3.py
```

Submission:

- Create a file named **`extra3.py`**
- Put your name in a comment at the top
- Submit it via CCLE

This extra credit assignment is worth an extra 4% towards your final grade.

Part 0 - Set Up

To use the Pymarc package you will need to install it using **`pip`**.

```
pip install pymarc
```

If you have both Python2 and Python3 installed (most Mac users) be sure to indicate that you want to install it for Python3 by using **`pip3`** instead.

```
pip3 install pymarc
```

Part 1 - Read the MARC data

- Use the **`sample_marc.mrc`** file as the source of data.
- Open the file using the `pymarc` library. See the week 7 lecture notes for an example of how to do this.
- Grab the following fields from each record:
    - The *title* from field **`245`** subfields **`a`** and **`b`**. You can use the **`title()`** convenience method instead of grabbing the individual subfields.
    - The *year published* from field **`260`** subfield **`c`**. You can use the **`pubyear()`** convenience method instead.
    - The *URL* for the digital copy from field **`856`**subfield **`u`**.
    - The *call number* from field **`090`**.
        - Once you grab the 090 field, use the**`format_field()`** method to convert the subfields into a nice single string:

          ```
          field = record['090']
          string = field.format_field()
          ```

        - *Beware!* Not all records have a call number. So test it before you call the`format_field()` method. If the record does not have a call number, use an empty string.

Part 2 - Save the Selected Data

- Now save the data for each record in a row in a CSV file like so:

```
title,pubyear,url,callnum
```

- Save the file as **`marc_data.csv`**