CIS 500 – Foundations of Software Practice
Winter 2023, Project #2

Zip Code ↔ Bar Code Converter
Due Date: Thursday, February 23rd, 2023

Objectives

- Functions
- Strings
- Exceptions
- Menu-driven user interface

Assignment

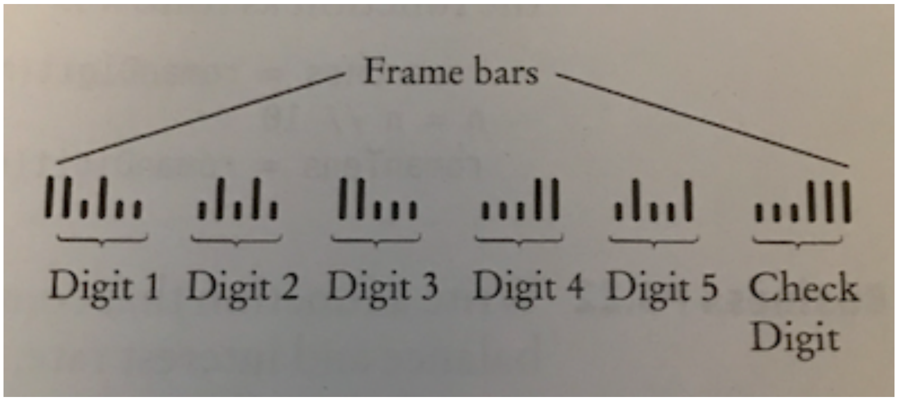
This assignment is loosely based on the problem "**Business P5.25:** ***Postal Bar Codes***" at the end of Chapter 5 in *Python for Everyone* textbook.

Your task is to implement functions defined in `project2.py` file that are incomplete. For each function that has a **`pass`** statement as its body, replace that statement with your implementation.

You will need to use the information in the table below. A vertical bar ("|") represents a full bar and a semicolon (":") represents a half bar in the postal bar code notation (See the figure on the next page).

Digit	Bar Code
0	||:::
1	:::||
2	::|:|
3	::||:
4	:|::|
5	:|:|:
6	:||::
7	|:::|
8	|::|:
9	|:|::

The encoding scheme for a five-digit code is shown in the figure below, which shows the postal bar code for zip code 95014. There are full-height frame bars on each side. The five encoded digits are followed by a check digit. The check digit is computed as follows: Add up all digits, and choose the check digit to make the sum a multiple of 10. For example, the zip code 95014 has a sum of 19, so the check digit is 1 to make the sum equal to 20.



Project Files

You are provided with two files for this project. Keep these files in a folder designated for this project. You may want to name this folder `Project2`.

- **`project2.py`:** Implement the following functions in this file:
  - `digit_to_code()`
  - `code_to_digit()`
  - `zipcode_to_barcode()`
  - `barcode_to_zipcode()`
  - `main()`

- **`project2_tests.py`**: This file contains unit tests for testing your implementation of the above functions in `project2.py` file. **DO NOT MODIFY THIS FILE.**

- My evaluation of your project will be based on the following two items (see grading rubric in `Project2_Rubric.pdf` file):
  - Implementation of `main()` to provide menu-driven user interface to use this program.
  - Number of unit tests from `project2_tests.py` file that pass.

Running Program from Command Line

Issue the following command at command prompt from inside the project directory:

```
$ python project2.py
```

The following is a sample execution of the program that shows few conversions:

```
$ python project2.py
Postal Zip Code <--> Postal Bar Code Converter Program

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 1
```

```
Eenter Postal Zip Code: 95014
The bar code for this zip code: ||:|:::|:|:||::::::||:|::|:::|||

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 1
Eenter Postal Zip Code: 49505
The bar code for this zip code: |:|::||:|:::|:|:||::::|:|:|:::||

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 1
Eenter Postal Zip Code: 49525
The bar code for this zip code: |:|::||:|:::|:|:::|:|:|:|::|:|:|

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 2
Eenter Postal Bar Code: ||:|:::|:|:||::::::||:|::|:::|||
The zip code for this bar code: 95014

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 2
Eenter Postal Bar Code: |:|::||:|:::|:|:||::::|:|:|:::||
The zip code for this bar code: 49505

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 2
Eenter Postal Bar Code: |:|::||:|:::|:|:::|:|:|:|::|:|:|
The zip code for this bar code: 49525

1. Zip Code to Bar Code Conversion
2. Bar Code to Zip Code Conversion
3. Quit Program

Enter your selection (1, 2, or 3): 3
Good Bye
```

Running Unit Tests

Issue the following command at command prompt from inside the project directory:

```
$ python project2_tests.py
```

Project Deliverables (VERY IMPORTANT)

1. Upload only **`project2.py`** file on Blackboard by end of day (11:59PM) on due date. Note the filename contains all lowercase letters.

2. Evaluation of your project will be based on the following two items (see the grading rubric in `Project2_Rubric.pdf` file):
   - Implementation of `main()` to provide menu-driven user interface to use this program.
   - Number of unit tests from `project2_tests.py` file that pass.

3. The submission time on Blackboard will be used as the official submission date/time.

4. **Late penalty (10% per day up to three days late max) applies after Thursday, February 23rd**.