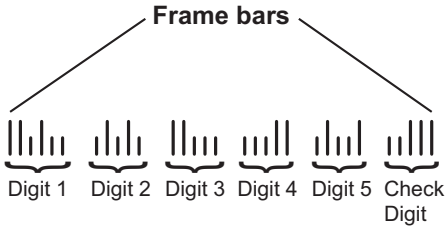
ECE-203 Programming for Engineers
Homework Week 4

For faster sorting of letters, the United States Postal Service encourages companies that send large volumes of mail to use a bar code denoting the zip code.

```
********************** ECRLOT ** CO57

CODE C671RTS2
JOHN DOE                              CO57
1009 FRANKLIN BLVD
SUNNYVALE       CA 95014-5143
```



The encoding scheme for a five-digit zip code is shown in the above figure. The five zipcode digits are encoded using five clusters of bars, where each bar is one of two heights: short or tall. The five encoded digits are followed by a check digit, which is computed as follows:

- Add up all the digits in the zipcode to make the sum Z
- Choose the check digit C so that $Z + C$ is a multiple of 10

For example, the zipcode 95014 has a sum of $Z = 9 + 5 + 0 + 1 + 4 = 19$, so the check digit C is 1 to make the total sum $Z + C$ equal to 20, which is a multiple of 10.

Each digit of the zipcode, as well as the check digit, is individually encoded according to the table below, where 0 denotes a short bar and 1 a tall bar:

Digit	Bar 1 (weight 7)	Bar 2 (weight 4)	Bar 3 (weight 2)	Bar 4 (weight 1)	Bar 5 (weight 0)
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	1	0	0	0	1
8	1	0	0	1	0
9	1	0	1	0	0
0	1	1	0	0	0

Decoding from the bar pattern to the original digit can be easily computed using the column weights 7, 4, 2, 1, and 0. For example, 01100 (short, tall, tall, short, short) decodes to:

$$0 \times 7 + 1 \times 4 + 1 \times 2 + 0 \times 1 + 0 \times 0 = 6$$

Unfortunately, the digit 0 is an exception. If you apply the same decoding algorithm for 0, you end up with:

$$1 \times 7 + 1 \times 4 + 0 \times 2 + 0 \times 1 + 0 \times 0 = 11$$

So, bar clusters that decode to 11 will require special handling since the true decoded value is actually 0.

Write a program that asks the user for a zip code and prints the bar code. Use `!` for tall bars and `.` for short bars. For example, if the user entered 95014, your program should produce:

```
!!.!..  .!.!.  !!...  ...!!  .!..!  ...!!!
```

Notice how there is an extra tall bar at the start and another at the end. These are called the frame bars and they are used to determine where the code starts and stops.

You will need to implement the following two functions:

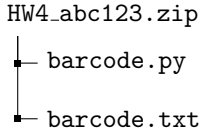
```
printDigit(digit)
printBarCode(zip_code)
```

which you will use to implement the encoding algorithm. `printDigit()` should accept a single parameter (representing a digit between 0 and 9) and produce a single "cluster" of bars. The `printBarCode()` function should accept a single parameter representing a full 5 digit zipcode and produce a full bar code of 5+1 clusters with frame bars. Each cluster should be crafted using `printDigit()`.

Name your program **barcode.py** and save its output as **barcode.txt**. Your output file should show your program testing the following zipcodes: 19104, 33139, 02201, 49726, 70802, 38103, 98104, and 49918.

Submitting Your Assignment

Create a zip file with the following structure:



name your zip file using your Drexel user id. For example: `HW3_abc123.zip`. Upload your zip file to `learn.drexel.edu` using the assignment submission link found on the course website.

Points will be deducted for improperly packaged submissions.