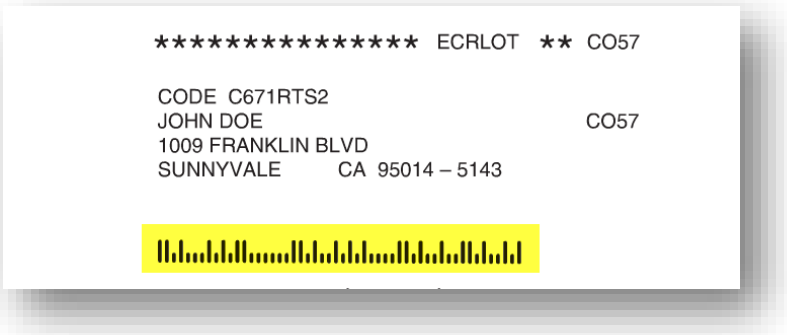
Homework

7

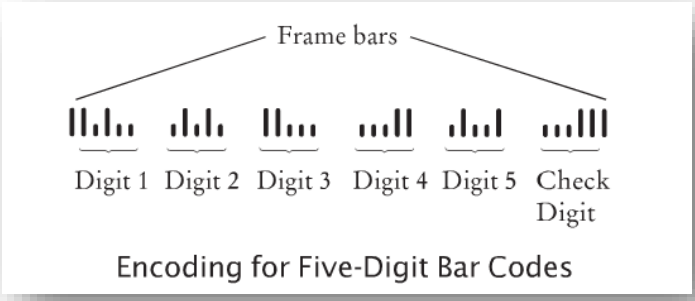
Going Postal

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:



The encoding scheme for a five-digit zip code is shown below.

- There are full-height **frame bars** on each side.
- The five encoded digits are followed by a **check digit**, which is computed by adding up all digits, and choose a check digit to make the sum a multiple of **10**.
- For example, the number **95014** has a sum of **19**, so the check digit is **1** to make the sum equal to **20**.



Encoding for Five-Digit Bar Codes

Each digit of the zip code, and the check digit, is encoded according to the following table where 0 denotes a **half bar** and 1 a **full 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

To solve this problem, write three functions (pay attention to the return types):

- `string codeForDigit(int digit)` which takes one digit and returns the appropriate code for that digit, using the table above. Use `':'` for the half-bars and `'|'` for the full bars.
- `int checkDigit(int zip)` which calculates the check digit according to the rules shown above.
- `string barCode(int zip)` which returns an entire bar code by breaking the number into individual digits, encoding that digit, and adding it to the `string` return value. Finally, calculate and encode the check digit, **surround the entire code with the frame bars**, and return it.

Here are the steps to solve this problem:

1. Write and document the prototypes the header file. Add header guards.
2. Stub out your functions in the implementation file. Run `make test` to make sure you have the mechanics correct before going on.
3. Implement the `codeForDigit()` function first. Run the tests for that.
4. Calculate the `checkDigit()` function and run the tests for that.
5. Finally, write and test the `barCode()` function (which will be fairly short).

A Few Hints

- `codeForDigit()` works great with just a `switch`.
- For `checkDigit()`, use a limit loop as well as the remainder operator to find the digit that gets the sum to the next multiple of `10`.
- For `barCode()`, first calculate the check digit. Extract each digit using a limit loop and get its code; use the building-a-string pattern to add the code to the result. Finally, add the code for the check digit and the two frame bars.

Use `make test` to test your code or `make run` to run any student tests. Once your score is OK, use `make submit` to turn it in.

If you get stuck, ask for help on the discussion board, or come by my office hours (early!!!).