

Objectives: Use functions, loops, and input/output streams

The US Postal Services uses bar codes to represent zip codes in order to sort them quickly using machines. The machine will read the bar code and direct the letter to the right location.

Write a C++ program that generate bar codes. The program should read the zip codes from an text file and determine the complete bar codes. All the codes along with the zip codes must be stored in another file.

The program must ask the user for both the input and output file names.

Your program should generate the codes repeatedly from the file until the end of the input file is reached. Your program should output an error message if the zip code read is invalid (for example, non-digit character, fewer or more than 5 digits). See sample output file below.

Bar Codes

The code is represented by a series of short and long bars. The following table shows the barcodes for each digit.

Value	Encoding
1	u l
2	u l
3	u l
4	u l
5	u l
6	u l
7	l l
8	l l
9	l l
0	l l

Use ':' and '|' to represent short and long bars.

A check digit is also added at the end of the barcode. The check digit is determined by totaling all the digits of the zip code and adding a digit that makes the sum divisible by 10. For example, if the sum of all the digits is 38 then the check digit is 2, if the sum is 40 the check digit is 0, etc.

A long bar is added at the beginning and at the end of the barcode (total of 32 bars). The following is an example of the code 45701 (sum = 17, check digit is 3)

	: ::	: ::	::	::	::	:: :	
	4	5	7	0	1	3	
						Check	
						Digit	

Your program must include at least the following two functions (**do not change the names or the prototypes**):

- `string getDigitCode (char digit):` return a barcode for a single digit
- `int getCheckDigitValue(int sum):` return the check digit

You are encouraged to add other functions.

Sample Input File (The sample file `zips.txt` is included with your repository):

```
45701
456
56-45
564565
12345
```

The sample file `zips.txt` is included with your repository.

Sample Output File:

```
45701      |:|::| |:|:| |:~:~| ||::: ~::|| ~:|~:~|
456        Error: zip code must be 5 digits
56-45      Error: Code is invalid
564565     Error: Code is invalid
12345      |:~:~| ~:~:~| ~:~|: |:~:~| |:~:~| ~:~:~|
```

Sample user interface:

```
Enter the input file name: zips.txt
Enter the output file name: codes.txt
The file codes.txt has been created
```

Grading:

Programs that contain syntax errors will earn zero points.

Programs that do not include functions other than main, will also earn zero points.

Programs that use global variables other than constants, will earn zero points.

Programs that use libraries NOT discussed in class will earn zero points.

Your grade will be determine using the following criteria:

- Correctness: the program works as requested above (35 points).
 - (5 points) The code is printed correctly (32 bars)
 - (20 Points) The required functions are implemented

- (10 points) Error checking
- **Documentation and Style (5 points)**

Follow the coding style outline on GitHub:

<https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md>