

CS 1160-01 Introduction to Computer Science and Programming Methods
Programming Assignment 2
Due, Monday, January 24, 2005

POSTNET encoded ZIP codes

The POSTNET encoding for ZIP codes involves a sequence of long and short bars. For this assignment, we will use “.” (period) to represent a short bar and “:” (colon) to represent a long bar. Each digit is represented by a series of 5 bars consisting of three short and two long bars. Mathematically the 5 “places” of the bar code can be thought of as representing the values 7,4,2,1,0, respectively. This means, explicitly, that the digits 0 to 9 are represented, respectively, by the strings “:....”, “...:”, “...:”, “...:”, “...:”, “...:”, “...:”, “:....”, “:....”, and “:....”. In addition to the encodings of the digits, the POSTNET code starts and ends with an additional long bar. This means, for example, that the ZIP 94542 could be encoded as:

:.....

In fact the POSTNET code for 94542 is really:

:.....

as if it were actually 945426.

The additional digit added is a *check digit* to help detect errors. The *check digit* is chosen so as to make the *sum of the digits* a multiple of 10.

Your task

You are to write a program which will accept as input a 5 digit ZIP code and will print out both the ZIP code entered and the POSTNET code corresponding to that ZIP code. [NOTE: If you want to extend your program to handle 9 digit ZIP+4 codes, that's OK.]

You will probably find it useful to define a number of “const strings” to represent the elements of the POSTNET code. Also, reading the ZIP code input as a string will make part of your task easier. [The alternative would be to read it as an int.]

Assuming that you have the ZIP code as a string, you will need to know that the integer codes assigned to the chars '0' to '9' are contiguous integers. Converting these character codes to actual integers involves a *converting the type* to int and using the *offset* from the code for '0'. This means, for example, that in order to convert the character '6' to an int one would do the calculation “int('6') - int('0')”. More generally, if n is a variable of type char whose current value is some digit character, then the int value represented by that digit is “int(n) - int('0')”.

Your program will involve small loops (for moving through the input string and computing the check-digit) and nested if-else branches.

You should, of course, *validate your input* and exit with an appropriate error messages in the situation in which the entered ZIP is not of a proper length or in which the input is not made up wholly of digits. [If you have extended your program to handle ZIP+4, it will be *your decision* as to whether to *allow* or *require* or *forbid* a separating hyphen, as in 94542-3092.]

Reminder: Don't forget the initial and final “:” on your POSTNET code.

What to turn in

As usual, you should turn in your source code, which should be in good style and well commented. You should also turn in the results of enough runs of your program to convince one that it works correctly.

Afterthought: If you would like to enhance your program from “single shot” capability to enable it to process multiple ZIP codes during each run, you could make use of the fact that the C++ library class string makes it possible to use the equality operator (==) to test equality of strings. Using this idea, you could allow your input string for ZIP code to accept a string like “quit” and then embed most of your program in a while loop that tests your input string against the constant string “quit”.