CSC 362 Programming Assignment #1
Due: Thursday, September 11

Write a program to input a text file, one character at a time, and decode the contents of the file, saving the decoded content to an output file. The text in the file will consist of three types of characters, real characters, dummy characters, and code characters. The real characters are to be copied to the output file. The code characters will tell you which characters are dummy characters. Code characters will consist of the letter 'q' followed by a digit, or the letter 'z' or the letter 'x' followed by any punctuation mark followed by a digit. The control letters (z, x, q) can appear in either upper or lower case. The digit will tell you how many dummy characters follow. The digit will be 0-9 only, nothing more than 9. If the digit is 0, then do not skip any characters. For instance, the text "Hz!3abciq2de tX%1fheq1 rZ^3g he" will be decoded as follows (underlined characters indicate real characters): H, z!3 (code to skip three letters), abc (dummy characters), i, q2 (code to skip two letters), de (dummy characters), space, t, X%1 (code for skipping one character), f (dummy character), he, q1 (code to skip one character), space (dummy character), r, Z^3 (code to skip 3 characters), gspaceh (dummy characters), e. So the text is "Hi there".

Your program needs to keep track of:

1. the number of original characters in the file (**not** including the final EOF)
2. the number of characters in the decoded file (**not** including the final EOF)
3. the total number of code sequences (the above example has 5: z!3, q2, X%1, q1, Z^3)

Output to the console window a report on the file decoded:

1. percentage decrease in the file size
2. total number of code sequences

The percentage decrease is computed as 100 * (input character - output characters / input characters), formatted to 2 decimal points of accuracy with a % following, as in 31.67%.

The <ctype.h> library contains useful character functions that you will find useful in this program. The function isdigit(c) can be used to determine if a character is a digit (0-9) or not. The function ispunct(c) can be used to determine if a character is a punctuation mark or not. Convert a char digit to a number using: `number = (int) c - 48;` Characters are stored as int values in ascii, and the character '0' is stored as 48. So for instance, '9' is stored using ascii value 57, and 57 - 48 = 9. NOTE: the digit can be 0 meaning that you should not skip any additional letters other than the code itself (e.g., q0 or x$0 would skip no additional letters).

Run your program on the three data files on the instructor's website. The output expected for input file 1 is shown below so that you can test your program. Hand in a printout of your program (commented) and the output (both the files and the console windows) when running your program on input files 2 and 3.

File 1 decoded: `Hi there, how are you doing today?`

Console output:
```
Percent decrease:      61.80%
Number of codes:       13
```