

## Programming Exercises

### 1. Initials

Write a program that gets a string containing a person's first, middle, and last names, and displays their first, middle, and last initials. For example, if the user enters John William Smith, the program should display J. W. S.

### 2. Sum of Digits in a String

Write a program that asks the user to enter a series of single-digit numbers with nothing separating them. The program should display the sum of all the single digit numbers in the string. For example, if the user enters 2514, the method should return 12, which is the sum of 2, 5, 1, and 4.

### 3. Date Printer

Write a program that reads a string from the user containing a date in the form mm/dd/yyyy. It should print the date in the format March 12, 2018.

### 4. Morse Code Converter

Morse code is a code where each letter of the English alphabet, each digit, and various punctuation characters are represented by a series of dots and dashes. Table 8-4 shows part of the code.

Write a program that asks the user to enter a string, then converts that string to Morse code.

**Table 8-4** Morse code

Character	Code	Character	Code	Character	Code	Character	Code
space	<i>space</i>	6	- . . .	G	- - .	Q	- - - .
comma	- - . - -	7	- - - . .	H	. . . .	R	. - .
period	. - . - -	8	- - - - .	I	..	S	... .
question mark	. . - - -	9	- - - - - .	J	. - - -	T	-
0	- - - - -	A	. -	K	- . -	U	. . -
1	. - - - -	B	- . . .	L	. - . .	V	. . . -
2	. . - - -	C	- . - .	M	--	W	. - -
3	. . . - -	D	- . .	N	- .	X	- . . -
4	. . . . -	E	.	O	- - -	Y	- . -
5	. . . . .	F	. . - .	P	. - - .	Z	- - - .

## 5. Alphabetic Telephone Number Translator

Many companies use telephone numbers like 555-GET-FOOD so the number is easier for their customers to remember. On a standard telephone, the alphabetic letters are mapped to numbers in the following fashion:

- A, B, and C = 2
- D, E, and F = 3
- G, H, and I = 4
- J, K, and L = 5
- M, N, and O = 6
- P, Q, R, and S = 7
- T, U, and V = 8
- W, X, Y, and Z = 9

Write a program that asks the user to enter a 10-character telephone number in the format XXX-XXX-XXXX. The application should display the telephone number with any alphabetic characters that appeared in the original translated to their numeric equivalent. For example, if the user enters 555-GET-FOOD, the application should display 555-438-3663.

## 6. Average Number of Words

If you have downloaded the source code from the Computer Science Portal you will find a file named `text.txt` in the *Chapter 08* folder. The text that is in the file is stored as one sentence per line. Write a program that reads the file's contents and calculates the average number of words per sentence.

(You can access the Computer Science Portal at [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis).)

## 7. Character Analysis

If you have downloaded the source code you will find a file named `text.txt` in the *Chapter 08* folder. Write a program that reads the file's contents and determines the following:

- The number of uppercase letters in the file
- The number of lowercase letters in the file
- The number of digits in the file
- The number of whitespace characters in the file

## 8. Sentence Capitalizer

Write a program with a function that accepts a string as an argument and returns a copy of the string with the first character of each sentence capitalized. For instance, if the argument is “hello. my name is Joe. what is your name?” the function should return the string “Hello. My name is Joe. What is your name?” The program should let the user enter a string and then pass it to the function. The modified string should be displayed.

## 9. Vowels and Consonants

Write a program with a function that accepts a string as an argument and returns the number of vowels that the string contains. The application should have another function

that accepts a string as an argument and returns the number of consonants that the string contains. The application should let the user enter a string, and should display the number of vowels and the number of consonants it contains.

### 10. Most Frequent Character

Write a program that lets the user enter a string and displays the character that appears most frequently in the string.

### 11. Word Separator

Write a program that accepts as input a sentence in which all of the words are run together, but the first character of each word is uppercase. Convert the sentence to a string in which the words are separated by spaces, and only the first word starts with an uppercase letter. For example the string “StopAndSmellTheRoses.” would be converted to “Stop and smell the roses.”

### 12. Pig Latin

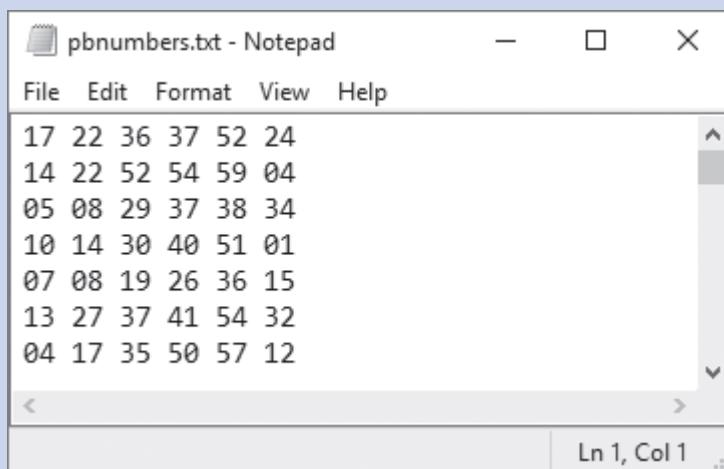
Write a program that accepts a sentence as input and converts each word to “Pig Latin.” In one version, to convert a word to Pig Latin, you remove the first letter and place that letter at the end of the word. Then, you append the string “ay” to the word. Here is an example:

English:	I SLEPT MOST OF THE NIGHT
Pig Latin:	IAY LEPTSAY OSTMAY FOAY HETAY IGHTNAY

### 13. PowerBall Lottery

To play the PowerBall lottery, you buy a ticket that has five numbers in the range of 1–69, and a “PowerBall” number in the range of 1–26. (You can pick the numbers yourself, or you can let the ticket machine randomly pick them for you.) Then, on a specified date, a winning set of numbers is randomly selected by a machine. If your first five numbers match the first five winning numbers in any order, and your PowerBall number matches the winning PowerBall number, then you win the jackpot, which is a very large amount of money. If your numbers match only some of the winning numbers, you win a lesser amount, depending on how many of the winning numbers you have matched.

In the student sample programs for this book, you will find a file named `pbnumbers.txt`, containing the winning PowerBall numbers that were selected between February 3, 2010 and May 11, 2016 (the file contains 654 sets of winning numbers). Figure 8-7 shows an example of the first few lines of the file’s contents. Each line in the file contains the set of six numbers that were selected on a given date. The numbers are separated by a space, and the last number in each line is the PowerBall number for that day. For example, the first line in the file shows the numbers for February 3, 2010, which were 17, 22, 36, 37, 52, and the PowerBall number 24.

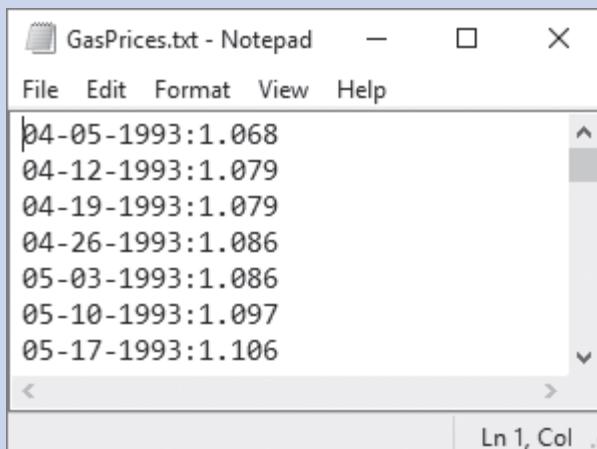
**Figure 8-7** The pbnumbers.txt file

Write one or more programs that work with this file to perform the following:

- Display the 10 most common numbers, ordered by frequency
- Display the 10 least common numbers, ordered by frequency
- Display the 10 most overdue numbers (numbers that haven't been drawn in a long time), ordered from most overdue to least overdue
- Display the frequency of each number 1–69, and the frequency of each Powerball number 1–26

#### 14. Gas Prices

In the student sample program files for this chapter, you will find a text file named `GasPrices.txt`. The file contains the weekly average prices for a gallon of gas in the United States, beginning on April 5th, 1993, and ending on August 26th, 2013. Figure 8-8 shows an example of the first few lines of the file's contents:

**Figure 8-8** The GasPrices.txt file

Each line in the file contains the average price for a gallon of gas on a specific date. Each line is formatted in the following way:

*MM-DD-YYYY:Price*

*MM* is the two-digit month, *DD* is the two-digit day, and *YYYY* is the four-digit year. *Price* is the average price per gallon of gas on the specified date.

For this assignment, you are to write one or more programs that read the contents of the file and perform the following calculations:

**Average Price Per Year:** Calculate the average price of gas per year, for each year in the file. (The file's data starts in April of 1993, and it ends in August 2013. Use the data that is present for the years 1993 and 2013.)

**Average Price Per Month:** Calculate the average price for each month in the file.

**Highest and Lowest Prices Per Year:** For each year in the file, determine the date and amount for the lowest price, and the highest price.

**List of Prices, Lowest to Highest:** Generate a text file that lists the dates and prices, sorted from the lowest price to the highest.

**List of Prices, Highest to lowest:** Generate a text file that lists the dates and prices, sorted from the highest price to the lowest.

You can write one program to perform all of these calculations, or you can write different programs, one for each calculation.