

IC210, Practice 6-Week Practicums

These practice problems are provided in order to give you more practice. They are intended to help you be as prepared as possible for the 6-week practicum exam.

TREAT THESE EXERCISES AS ACTUAL PRACTICUMS. DO NOT READ THE PROBLEM UNTIL YOU ARE READY TO COMPLETE IT.

It is recommended that you attempt to recreate the testing environment as close as possible to real conditions as possible. In order to get the most benefit from these exercises you should limit yourself to the following conditions:

- Time: 50 minutes
- Resources: Only your homework, labs, paper-based notes, the official class notes from the web, and your textbook.
- Collaborating with someone else is not recommended. It will only hurt you in the long run.

Some tips for Practicum (and general programming) Success

- Start with a pen/pencil and paper. Take the first five to ten minutes of the test period to read the problem carefully and write down an outline or flowchart highlighting the major steps and obstacles of the problem.
- Before writing the substantive code, transfer your outline into your program by using comments. These comments will serve as guideposts as you write your program, and if nothing else will provide your instructor with some idea as to what your intentions were.
- Focus on one step at a time. Don't try to do too much at once.
- Note that the sample input is just that, *sample input*. It is not the only input that your program should be able to handle. Make sure you carefully read the specified input and your program can handle all variations of **valid** input.
- Unless otherwise specified, assume that the user will not purposefully provide invalid input. If the program specifies integer values, the user will not input floating point numbers.
- Keep it simple. Don't try to be too fancy in developing your program. Use the simplest C++ constructs and don't outsmart your self.
- Compile often. Don't go too far before you compile and fix any errors.
- No matter what, make sure your turn in a program that compiles!!

Remember:

- The Practicum is intended to test your knowledge of C++ to this point. While the problem requires thought and planning, it is a straight forward problem that only requires the skills you've learned to this point.
- You are being graded on function, not style. Proper indention is required (so you don't confuse yourself or your instructor). Comments are highly recommended but not required.

STOP – Don't turn the page until you are ready to actually practice a practicum!

Problem 1:

Write a C++ program called `YourLastNamePractice1.cpp` (e.g. `SmithPractice1.cpp`) that first prints out your name and alphacode.

Your program is to then read a sequence of letters and spaces, from the standard input (the keyboard, not a file) and output to the screen:

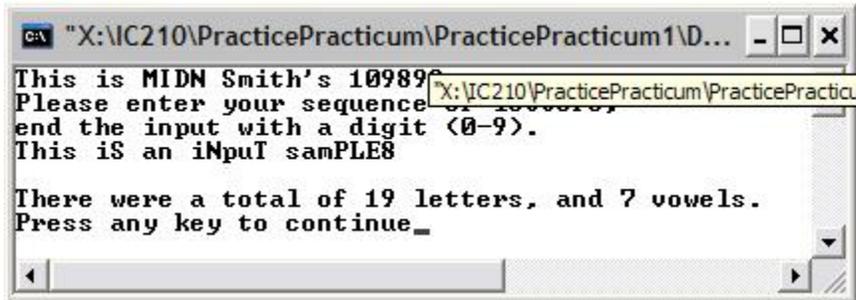
1. The total number of letters
 - There can be spaces but they are not letters and will not count towards the total
2. The total number of vowels

Your program should stop reading **as soon as it sees a number**

For instance, if the user entered:

This iS an iNpuT samPLEx

the output should look approximately like this:



```
C:\ "X:\IC210\PracticePracticum\PracticePracticum1\D...
This is MIDN Smith's 10989
Please enter your sequence
end the input with a digit <0-9>.
This iS an iNpuT samPLEx

There were a total of 19 letters, and 7 vowels.
Press any key to continue_
```

Problem 2: (easier than a standard practicum)

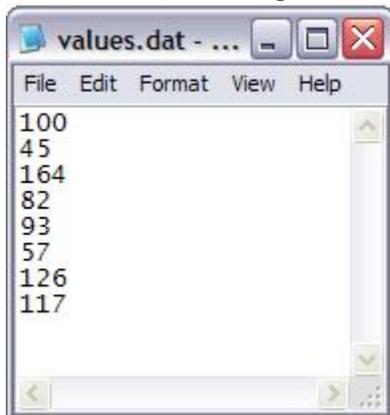
Write a C++ program called YourLastNamePractice2.cpp (e.g. SmithPractice2.cpp) that first prints out your name and alphacode.

Your program is to then read in a series of numbers from a file specified by the user and output to the screen:

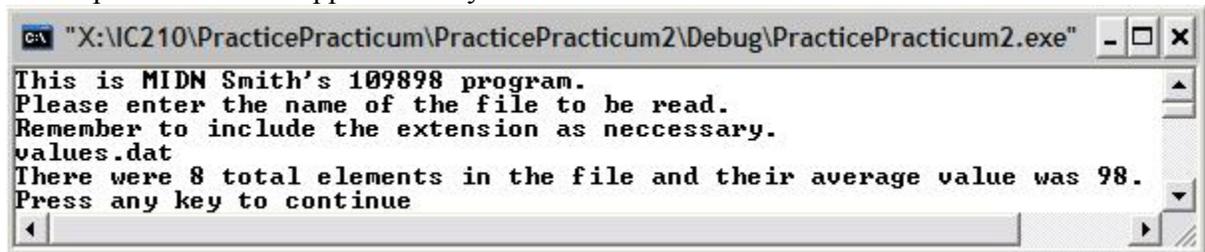
1. The total number of elements read
2. The file's average

Your program should read the entire file.

For instance, if the input file looked like this:



the output should look approximately like this:



Problem 3:

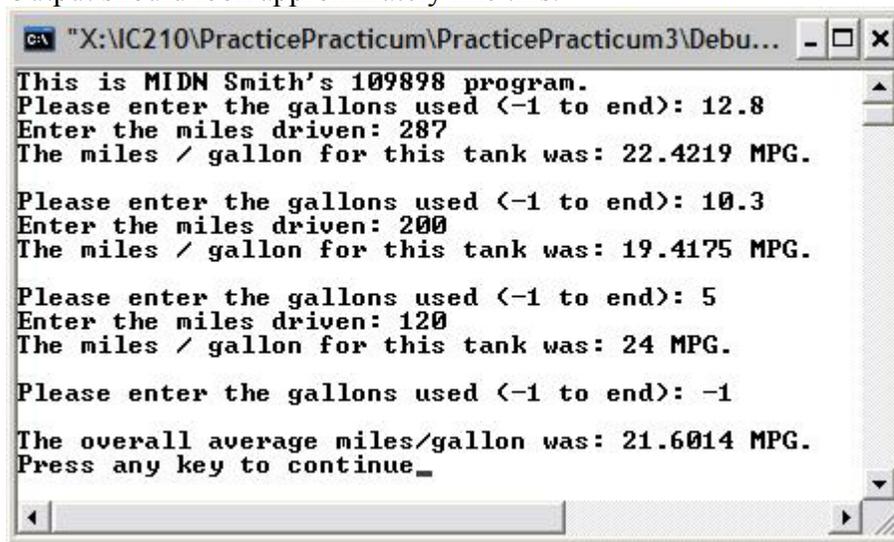
Because of the high price of gasoline, drivers are concerned with the mileage obtained by their automobiles. One driver has kept track of several tanks of gasoline by recording the miles driven and gallons used between each fill up.

Write a C++ program called YourLastNamePractice3.cpp (e.g. SmithPractice3.cpp) that first prints out your name and alphacode.

Your program is then to get the miles driven and gallons used from the user for each tankful. The program should calculate and display the miles per gallon obtained for each tankful. After processing all the information, the program should calculate and print the combined miles per gallon for all tankfuls.

The user will input **-1** to signal that he is done with input.

Output should look approximately like this:



```
C:\> "X:\IC210\PracticePracticum\PracticePracticum3\Debu...
This is MIDN Smith's 109898 program.
Please enter the gallons used (-1 to end): 12.8
Enter the miles driven: 287
The miles / gallon for this tank was: 22.4219 MPG.

Please enter the gallons used (-1 to end): 10.3
Enter the miles driven: 200
The miles / gallon for this tank was: 19.4175 MPG.

Please enter the gallons used (-1 to end): 5
Enter the miles driven: 120
The miles / gallon for this tank was: 24 MPG.

Please enter the gallons used (-1 to end): -1

The overall average miles/gallon was: 21.6014 MPG.
Press any key to continue_
```

Problem 4: (more challenging)

Write a C++ program called YourLastNamePractice4.cpp (e.g. SmithPractice4.cpp) that first prints out your name and alphacode.

Your program is to then prompt the user for an integer, **n**, and then read in a series of letters terminated by a number, and output to the screen:

- The sequence of letters with every **n**th switched from upper-case to lower- and vice versa as appropriate.

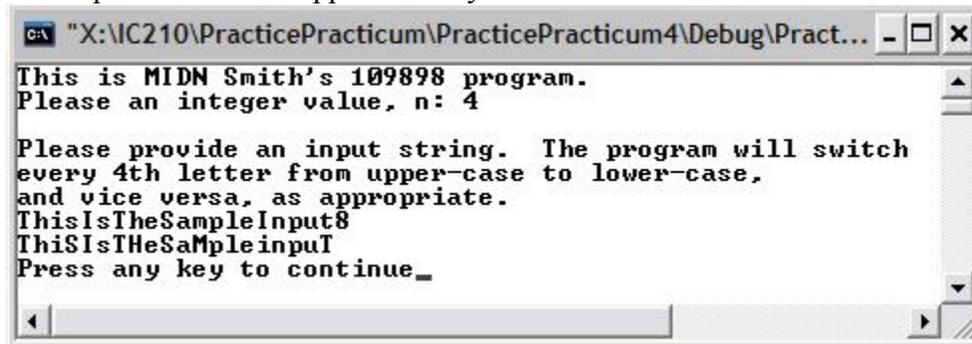
Your program should stop reading **as soon as it sees a number**

For instance, if the user's input was:

4

ThisIsTheSampleInput8

the output should look approximately like this:



```
C:\> "X:\IC210\PracticePracticum\PracticePracticum4\Debug\Pract...
This is MIDN Smith's 109898 program.
Please an integer value, n: 4

Please provide an input string. The program will switch
every 4th letter from upper-case to lower-case,
and vice versa, as appropriate.
ThisIsTheSampleInput8
ThisISTheSaMpleinput
Press any key to continue_
```

Problem 5:

Suppose you can buy a chocolate bar from the vending machine for \$1 each. Inside every chocolate bar is a coupon. You can redeem seven coupons for one bar from the machine.

For example, if you have 20 dollars then you can initially buy 20 chocolate bars. Each has a coupon, for a total of 20 coupons. You can redeem 14 coupons for two additional chocolate bars. These two additional chocolate bars give you two more coupons, so you now have eight coupons. This gives you enough to redeem for one final chocolate bar. As a result you now have 23 chocolate bars and two leftover coupons.

Write a C++ program called `YourLastNamePractice5.cpp` (e.g. `SmithPractice5.cpp`) that first prints out your name and alphacode.

Your program will then prompt the user for how many dollars to be spent and output to the screen:

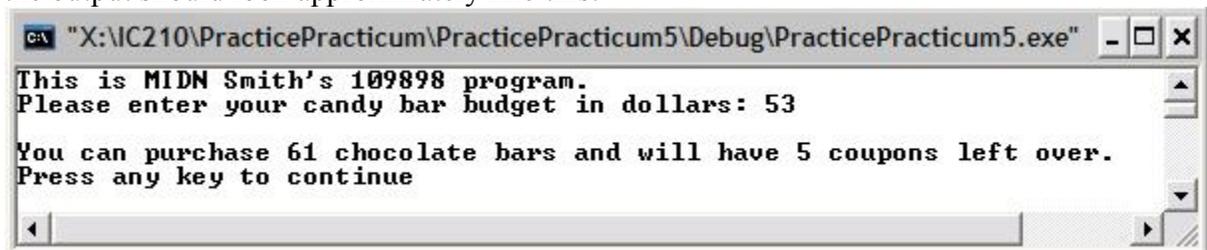
1. The total number of chocolate bars you can collect after spending all the money and redeeming as many coupons as possible.
2. The number of leftover coupons.

Use a loop to solve this problem. Don't use clever math tricks to solve it.

For instance, if the user's input was:

53

the output should look approximately like this:



```
"X:\C210\PracticePracticum\PracticePracticum5\Debug\PracticePracticum5.exe"
This is MIDN Smith's 109898 program.
Please enter your candy bar budget in dollars: 53
You can purchase 61 chocolate bars and will have 5 coupons left over.
Press any key to continue
```