Mr. Bellavia

EXERCISE 03 - MORE LOOP EXAMPLES

IMPORTANT: Before submission, make a copy of your **'Program.cs'** file for each question and then rename each file to the following:

File Names:

- *last_name_first name*_U3_E03_1_2.cs
- *last_name_first name*_U3_E03_3.cs
- *last_name_first name*_U3_E03_4.cs
- *last_name_first name*_U3_E03_5.cs
- *last_name_first name*_U3_E03_6.cs

Note: Along with last name and first name, make sure the end of the filename (i.e., before the **.cs**) has the **unit number**, **exercise number**, and **question number.** For example:

smith_*john*_***U1***_***E03***_**2**.*cs*

SUBMIT <u>1 FILE</u> FOR QUESTIONS 1 & 2:

1. Without using any built-in string functions, write a program that asks the user for a string **'s'** and a char **'c'** then counts how many times **'c'** appears in **'s'.**

2. Without using any built-in string functions, write a program that asks the user for a string and then outputs every second character of that string (**Hint:** Make use of the **modulus %** operator).

---

3. Without using any built-in string functions (except for .Length()), write a program that will ask the user for a string and counts how many times the same letter appears side by side. For example, consider the string:

"Hello ICS, what a magnificent school SJB is!"

This string contains the word 'Hello' in which the letter 'l' appears side by side and contains the word 'school' in which the letter 'o' appears side by side. Therefore, the number of side-by-side letters in this string is 2.

Hint: You will need create an extra **char** variable that holds the previous letter.

4. Without using any built-in string functions (except for .Length()), ask the user for a string and output that string with all vowels removed. For example:

Sample **Input** & **Output**:

```
Enter a string: Hello there world
String with no vowels: Hll thr wrld
```

Hint: On each iteration of your for-loop, append all the wanted characters to a new string.

5. Re-write the input/output files example from the lesson so that it can read an input file that is a comma delimited file **(input.csv).** A **comma delimited file** contains records (lines) where each record contains fields separated by a comma. Therefore, your input file should be changed to look like following:

Input file (input.csv):

```
Smith,John,87
Doe,Jane,93
Sacramanto,Bob,57
```

6. Write a program that asks the user for the number of records they wish to create, then ask for each record. Each record contains the following fields: **first name, last name, phone number,** and **email**

Sample **Input** & **Output**:

```
How many records do you wish to input: 5

Enter first name for record 1: John
Enter last name for record 1: Smith
Enter phone number for record 1: 9055763030
Enter email for record 1: smithjohn@hotmail.com

Enter first name for record 2: Doe
Enter last name for record 2: Jane
Enter phone number for record 2: 905587447
Enter email for record 2: doejane@gmail.com

(...and so on...)
```

Have your program generate a **comma delimited** output file where each record has the following field order:

```
last name, first name, phone number, email
```

For example:

Output file (output.csv):

```
Smith,John,9055763030,smithjohn@hotmail.com
Doe,Jane,905587447,doejane@gmail.com

(...and so on...)
```

Note: This question should be done using a **definite** loop!