Mr. Bellavia

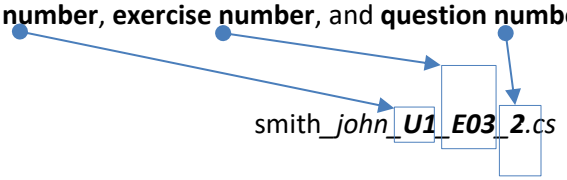
EXERCISE 02 - FOR-LOOPS

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_E02_1.cs
- *last_name_first name*_U3_E02_2.cs
- *last_name_first name*_U3_E02_3.cs
- *last_name_first name*_U3_E02_4.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:



IMPORTANT: Make use of **for-loops** for **all** the following questions:

1. Write a program using a **for-loop** that outputs:

a) the first ten **10** natural numbers each on a separate line:

```
1
2
3
4
5
6
7
8
9
10
```

b) every 3rd number from **0** to **21** each separated by a space on the same line:

```
0 3 6 9 12 15 18 21
```

c) every 4th number backwards from **50** to **40** each separated by a space on the same line:

```
50 46 42
```

d) All numbers between **'x'** (user input) and **35** each separated by a space on the same line with the sum of all those numbers printed at the end, for example:

```
Please enter an integer: 30
30 31 32 33 34 35
Sum = 195
```

Mr. Bellavia

2. Create a **for-loop** that cycles from **0** up to and including **20**. On each iteration, output the counter variable followed by a space, then output an asterisk (*) if the value of the for-loop counter variable is divisible by 2, an at symbol (@) if divisible by 3, and a dollar sign ($) if divisible by 4. The output of each iteration should be on a separate line. **Note:** Certain iterations may be divisible by more than one number. In such cases, you will have more than one symbol printed and separated by a space on that line. Sample output:

```
0 * @ $
1
2 *
3 @
4 * $
5
6 * @
7
8 * $
9 @
10 *
11
12 * @ $
13
14 *
15 @
16 * $
17
18 * @
19
20 * $
```

Hint: Make use of the modulus operator **'%'** (like we did in previous exercises) to determine if a number is divisible by 2, 3, or 4.

Bonus: If a number is **not** divisible by 2, 3, or 4 then output an exclamation mark (!). **Hint:** Make use of a Boolean flag.

Mr. Bellavia

*****For the next two questions (4 & 5) feel free to search and check out sample code on the internet!**

3. A prime number is a positive integer (i.e., natural number) that is **only** divisible by itself and 1. For example: 2, 5 & 11 are all prime numbers.

Write a program that determines if a user's input is prime or not.

4. In mathematics, the factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n (Wikipedia). For example:

5! = 5 x 4 x 3 x 2 x 1 = 120

The only stipulation is that 0! = 1

a) Write a program that asks the user for an integer **greater than 0** and then uses a for-loop to calculate the factorial of that positive integer. Your output should look like this:

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

```
Enter an integer number greater than 0: 5

5! = 120
```

Ensure that your program allows the user to type an integer **greater or equal to 0**. Remember that the factorial of '0' is a special case which equals 1:

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

```
Enter an integer number greater than 0: 5

0! = 1
```

b) Modify your program so that the output looks like the following:

```
Enter an integer number greater than 0: 5

5! = 5 x 4 x 3 x 2 x 1 = 120
```

Bonus: Notice that there is no 'x' to the right of the last number '1'