Project 4-1: Display a table of powers

Console

```
Welcome to the Squares and Cubes table

Enter an integer: 9

Number  Squared Cubed
======  ======= =====
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729

Continue? (y/n): y

Enter an integer: 3

Number  Squared Cubed
======  ======= =====
1       1       1
2       4       8
3       9       27

Continue? (y/n): n
```

Operation

- The application prompts the user to enter an integer.
- The application displays a table of squares and cubes from 1 to the value entered by the user.
- The application prompts the user to continue.

Specifications

- The formulas for calculating squares and cubes are:

```
square = x * x
cube = x * x * x
```

- Assume that the user will enter a valid integer.
- The application should continue only if the user enters "y" or "Y" to continue.

Project 4-2: Calculate the factorial of a number

Console

```
Welcome to the Factorial Calculator

Enter an integer that's greater than 0 and less than 10: 3
The factorial of 3 is 6.

Continue? (y/n): y

Enter an integer that's greater than 0 and less than 10: 4
The factorial of 4 is 24.

Continue? (y/n): y

Enter an integer that's greater than 0 and less than 10: 9
The factorial of 9 is 362880.

Continue? (y/n): n
```

Operation

- The application prompts the user to enter an integer from 1 to 9.
- The application displays the factorial of the number entered by the user.
- The application prompts the user to continue.

Specifications

- The exclamation point is used to identify a factorial. For example, the factorial of the number n is denoted by n!. Here's how you calculate the factorial of the numbers 1 through 5:

```
1! = 1                  which equals 1
2! = 1 * 2              which equals 2
3! = 1 * 2 * 3          which equals 6
4! = 1 * 2 * 3 * 4      which equals 24
5! = 1 * 2 * 3 * 4 * 5  which equals 120
```

- Use a for loop to calculate the factorial.
- Assume that the user will enter valid numeric data for the length and width.
- Use the long type to store the factorial.
- The application should continue only if the user enters "y" or "Y" to continue.

Enhancement

- Test the application and find the integer for the highest factorial that can be accurately calculated by this application. Then, modify the prompt so it prompts the user for a number from 1 to the highest integer that returns an accurate factorial calculation.

Project 4-3: Find the greatest common divisor of two positive integers

Console

```
Greatest Common Divisor Finder

Enter first number: 12
Enter second number: 8
Greatest common divisor: 4

Continue? (y/n): y

Enter first number: 77
Enter second number: 33
Greatest common divisor: 11

Continue? (y/n): y

Enter first number: 441
Enter second number: 252
Greatest common divisor: 63

Continue? (y/n): n
```

Operation

- The application prompts the user to enter two numbers.
- The application displays the greatest common divisor of the two numbers.
- The application prompts the user to continue.

Specifications

- The formula for finding the greatest common divisor of two positive integers x and y follows the Euclidean algorithm as follows:

  1. Subtract x from y repeatedly until y < x.
  2. Swap the values of x and y.
  3. Repeat steps 1 and 2 until x = 0.
  4. y is the greatest common divisor of the two numbers.

- Place the calculation for finding the divisor in a static method. You can use one loop for step 1 of the algorithm nested within a second loop for step 3.
- Assume that the user will enter valid integers for both numbers.
- The application should continue only if the user enters "y" or "Y" to continue.