Project 6

due at midnight on Mon 27 Oct (60 points)

The purpose of this assignment is to practice using **loops** (chapter 4). You will write (yet another) distance conversion program, but this time the user can specify the parameters of a **conversion table** – what value it starts with, where it ends, and how much to increase each row.

Sample runs

Below are some sample runs:

```
Enter start: 1.5
Enter stop: 2.5
Enter step: 0.2

Miles   Kilometers
 1.500   2.414
 1.700   2.736
 1.900   3.058
 2.100   3.380
 2.300   3.701
```

Another run:

```
Enter start: 10.2
Enter stop: 20
Enter step: 3

Miles   Kilometers
 10.200   16.415
 13.200   21.243
 16.200   26.071
 19.200   30.899
```

Alignment

In order to get a well-aligned table, we need to tell `cout` to use a fixed number of places after the decimal point. Otherwise it might print `4` miles in one row, then `4.25` in the next and `4.5` in the next, and the kilometers column would become unaligned:

```
Miles   Kilometers
 4   6.43736
 4.25   6.83969
 4.5   7.24203
 4.75   7.64436
 5   8.0467
 5.25   8.44904
 5.5   8.85137
```

To fix this, add this new library at the top of your program:

```
#include <iomanip>
```

and then use this statement somewhere before you start printing the table:

```
    cout << fixed << setprecision(3);
```

Here's what that same table looks like now:

```
Miles   Kilometers
 4.000   6.437
 4.250   6.840
 4.500   7.242
 4.750   7.644
 5.000   8.047
 5.250   8.449
 5.500   8.851
```

Error checking

You should do some simple error-checking:

```
Enter start: 19
Enter stop: 0
ERROR: stop must be greater than start.

Enter start: 7
Enter stop: 24
Enter step: 0
ERROR: step must be greater than zero.
```