1 Pretty Formatting

You'll learn more about `printf` and `scanf`—and how they interact with a loop.

To get you started, a portion of the solution is provided for you below; you must format and print the input to complete the solution.

1.1 Input Format

Every line of input will contain a string followed by an integer. Each string will have a maximum of 10 alphabetic characters, and each integer will be in the inclusive range from to 0 to 1,000.

1.2 Output Format

In each line of output there should be two columns:

- The first column contains the string and is left justified using exactly 15 characters.
- The second column contains the integer, expressed in exactly 3 digits; if the original input has less than three digits, you must pad your output's leading digits with zeroes.

Sample Input:

```
java 100
c 65
python 50
```

Sample Output:

```
================================
java            100
c               065
python          050
================================
```

Starter Code:

2 Table of Conversion

Write a program that will display a conversion table between Celcius and Fahrenheit. Ask the user (from `stdin`) for a lower limit in Celcius (`lb`) and an upper limit in Celcius (`ub`). Display a table of temperature between `lb` and `ub` in increments of 5.

Example

As an example:

```
lower limit: 31
upper limit: 46

Celcius         Fahrenheit
=========================
     31               87.8
     36               96.8
```

```
      41                105.8
      46                114.8
-------------------------
```

3 Command-Line Arguments

Read more online about `argc` and `argv`. Assuming that the command-line arguments are all integers, write a program that sums up the command-line arguments and prints that to the screen. For example, if your program is called `sum_all`, then `./sum_all 3 2 1` should display 6 on the screen.