Project: Program **p01** converts a positive, rational, decimal number to a corresponding value in a foreign base. A foreign base, f is constrained to the range $2 \leq f \leq 36$.

Program Files: Project **1** consists of file **p01.cpp**

File	Description
p01.cpp	File **p01.cpp** contains functions that process command line arguments and direct the process of reading, converting, and printing original input values together with corresponding output values.

Project file directory: **Project files must be stored in the root directory of your student account.** Failure to store project files in the root directory of your student account will result in a score of ***zero*** (***0***) for this project.

Command Line: Project **1** can be invoked with zero, one, or two program parameters. The first program parameter is the input file name. The second parameter is the output file name. Sample command lines together with corresponding actions by program **p01** are shown below. Boldfaced type indicates data entered at the keyboard by the user.

$ **p01**
Enter the input file name: **i01.dat**
Enter the output file name: **o01.dat**

$ **p01 i01.dat**
Enter the output file name: **o01.dat**

$ **p01 i01.dat o01.dat**

Input File: The input file contains pairs. The first value in the pair is a rational, decimal number. The second value in the pair is an integer specifying the foreign base f, where $2 \leq f \leq 36$. As an example, consider the following pairs:

```
10.375  2
10.375 16
```

Input File Errors: Your project must account for errors in the input file.

1. Your project must detect and report an input file containing an odd number of values.
2. Your project must detect and report an invalid input value to be converted.
3. Your project must detect and report an invalid foreign base.

Output File Specification: One line is printed for each pair in the input file. Three values are printed on a line. The first two values are the values read from the input file. The last value is the decimal equivalent of the input values.

Output File Format:

Each line is formatted. A single space is printed between each of the three values printed. The first value, a floating-point value, is printed right-justified in a field of twenty (20) spaces in fixed format with eight (8) fractional digits. The next value, an integer, is printed right-justified in a field of two (2) spaces. The last value, a string, is printed right-justified in a field of twenty (20) spaces. Four (4) factional digits are printed in the converted value.

Example:

```
          10.37500000  2              1010.0110
       11181.50000000 16              2bad.8000
```