**ENGR1100 – Fall 2018**

**Homework 11.2**

**Individual Assignment:** See the course syllabus for a definition of what constitutes an individual assignment.

**Task 1 (of 2)**

**Objective:** Apply knowledge of Python’s collection data types and conditional statements.

Suppose you have graduated from UC and are an engineer for Really Expensive Optics, Inc. You have been given the task of modeling a system of optical media to determine how light entering a lens system will behave. You have decided to code a proof-of-concept model for a ray-tracing program using your extensive knowledge of Python (with its corresponding flow diagram of course). Since it is only a proof‑of‑concept, for now you only have to deal with one ray of light and two optical media that are not air. In order to simplify your calculations, you have decided to use only the most basic concepts of Snell’s Law.

Looking back to your PHYS 2002 notes, you recall that Snell’s Law states that the behavior of a ray of light as it passes from one optical medium to another is determined from the media’s indices of refraction (*n*1 and *n*2) and the angles with which the light enters or leaves each medium. The relationship is given by the following equation from Snell’s Law:

 (Equation 1)

Since you are working with a ray of light that starts in medium 1 and passes through two optical media, you decide to draw a new diagram, similar to the one in your notes but more representative of your current problem. You come up with something like this:

*n*1

*n*2

*θ*2

*d*1

*d*2

*d*3

*θ*1

*Figure 1. Diagram of optical media.*

Relating this to Snell’s Law, you see that each angle exiting an optical interface (the point where one medium meets another) can be found from Equation 1, as long as you know the incoming angle and the two indices of refraction. Additionally, you notice immediately a situation in which an error (hint: Arithmetic error) could occur in finding the angle leaving the interface. This situation appears when light leaving a medium with higher refractive index into a medium with lower refractive index, which might be reflected back instead of refracting through the second medium when the incoming angle reaches a certain value (The largest incoming angle before this error occurs is called the *critical angle*), this situation is defined as total internal reflection.

For your application, you need to be able to handle one optical interface with two media. Additionally, you need to be able to determine the ending distance (shown as *d*3 in Figure 1) where the light ray will fall.

For inputs, you will prompt the user to enter (in this order):

1. Indices of refraction for the two media. Input them as lists or tuples (i.e one line input).
2. Incoming angle θ1 in degrees (1 value).
3. Vertical distances through the media from the point where the light ray starts (*d*1, *d*2, in Figure 1). Again, these will be input as lists or tuples.

After entering the values, you need to check them for obvious errors. Next, calculate and output the final horizontal distance (*d*3 in Figure 1). If the ray will not pass between media (i.e., it gets reflected), your program should display an error and terminate. Your file should be named **Homework\_11\_2\_Task1\_*UCusername*.py**. Two example outputs and test cases are given below to help you.

Enter indices of refraction for bottom two media:

1.26 1.33

Enter angle of incidence (in degrees):

48

Enter d1 and d2 (units):

1 1

Ending distance is:

2.102 units

Enter indices of refraction for bottom two media:

2 1

Enter angle of incidence (in degrees):

40

Enter d1, d2 (units):

1.25 2.3

Error, no refraction in the second media

**Task 2 (of 2)**

**Objective:** To practice how to take information from the user in the form of lists and utilize them in basic calculations and conditions.

In Homework 8.1, you created two Python functions that calculated the equivalent resistance of two resistors connected in either series or in parallel. After doing a little exploration, you’ve found that resistors connected in series and parallel have a unique purpose in circuit design. When two resistors are connected in series, they divide the total voltage into two smaller voltages. Similarly, when two resistors are connected in parallel, they divide the total current into two smaller currents. These types of applications are useful if, for instance, you have a 9V battery but only want 5V to run your circuitry.

In your research, you’ve found the relationships below for series and parallel connected resistors.

|  |  |
| --- | --- |
| **Series** | **Parallel** |
|  |  |

To help you the next time you need to design a circuit, you decide a write a Python program that will tell you the performance of two resistors connected in either series or parallel. Your program should prompt the user for:

* Two resistor values, entered as a list or tuple
* The value of the voltage source driving the circuit
* Whether the resistors are connected in parallel or series

Your program should check to ensure that the resistor values are appropriate (i.e. non-negative). If there is a problem with the resistor values, the program should display a message to the user and terminate. Otherwise, it should compute and display the following, depending on the way the resistors are connected:

* The total combined resistance (hint: use your functions from before!)
* The voltage on each resistor
* The current through each resistor

Name your file **Homework\_11\_2\_Task2\_*UCusername*.py**

**Submit the file(s):**

**Homework\_11\_2\_Task1\_*UCusername*.py**

**Homework\_11\_2\_Task2\_*UCusername*.py**

**to the appropriate location on Blackboard.**