ENGR 14100
Python 2 PA

Individual Assignment: See the course syllabus for a definition of what this constitutes.

Task 1 (of 2)

Learning Objectives: Create and implement user-defined functions in Python; Construct logic operations using comparison, membership, and logical operators to generate control flow statements in Python; Utilize conditional if-elif-else statements while programming in Python, as well as previously learned objectives such as: Output data from a function to the screen in Python; Apply course code standard in development of Python scripts; Modularize and comment code in Python for readability and reusability.

Imagine that you have graduated from Purdue and are now working as an engineer for Really Expensive Optics, Inc. You have been given a task for modeling a simple light control system using experimental optical media for a new medical imaging method. The model will determine the behavior of light passing through two optical media. Your team needs you to code a proof-of-concept model for a ray-tracing program using Python and develop a corresponding flow diagram as a working document for your team to use to understand the program. The model will later be integrated within a larger Python program modeling the imaging behavior. For this proof-of-concept, 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 241 notes, you recall that Snell's Law states that the behavior of a ray of light as it passes through to another medium is a function of the indices of refraction of the media (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:

$$n_1 sin\theta_1 = n_2 sin\theta_2 \qquad \text{(Equation 1)}$$

You draw a diagram indicating a ray of light that starts in medium 1 and passes through two optical media. You come up with something like this:

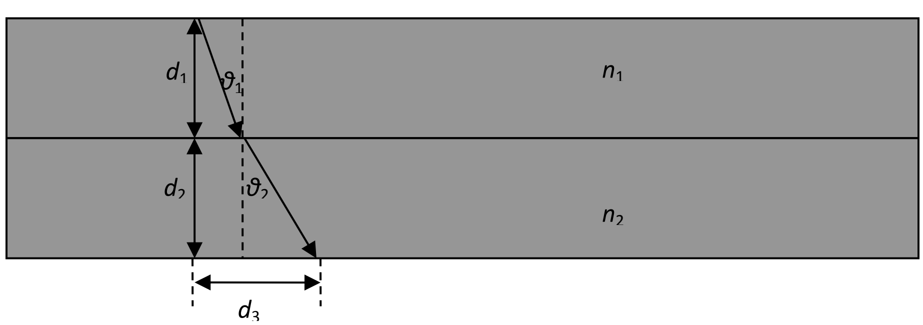

Figure 1. Diagram of optical media.

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 leaves a medium with a higher refractive index into a medium with a 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 as described in Figure 1. You need to determine if there is refraction or total internal reflection assuming n_1 and n_2 are given and θ_1 is an input. Based on this determination, your program should display a message indicating the behavior of the light ray. Also, when appropriate, the program should output the second angle θ_2, the ending distance d_3 where the light ray will fall (assuming d_1 and d_2 are given), and the critical angle.

Draw a flowchart (save in a PDF file) and write a Python program to perform this task.

You must make use of <u>two</u> different user-defined functions: one for calculating the angle θ_2 and another to calculate the distance d_3.

Example: Assuming the following values: $\theta_1 = 20°$, $n_1 = 1.7$, $n_2 = 1.3$, $d_1 = 5.3cm$ and $d_2 = 7.6cm$

The outputs of your flowchart and your Python program should give the following message:

```
Input incoming angle: 20.0
There is refraction with a leaving angle of 26.6 degrees
The ending distance for the light ray is 5.7 cm
For these two media, the critical angle is 49.9 degrees
```

Test your code with the two cases listed below. Save the results in the same PDF file you created. In your program, make sure to change parameters back to the ones defined in the example above before submitting files to BbL.

Case 1: $\theta_1 = 48°$, $n_1 = 1.26$, $n_2 = 1.33$, $d_1 = 1.0\ cm$ and $d_2 = 1.0\ cm$.

Case 2: $\theta_1 = 40°$, $n_1 = 1.90$, $n_2 = 1.05$, $d_1 = 1.25\ cm$ and $d_2 = 2.3\ cm$.

Task 1 Files:

1) Py2_PA_Task1_*login*.py
2) Py2_PA_*login*.pdf

Task 2 (of 2)

Learning Objectives: Create and implement user-defined functions in Python; Construct logic operations using comparison, membership, and logical operators to generate control flow statements in Python; Utilize conditional if-elif-else statements while programming in Python; as well as previously learned objectives such as: Output data from a function to the screen in Python; Apply course code standard in development of Python scripts; Modularize and comment code in Python for readability and reusability.

At your first co-op rotation with Ford Labs, you work on a vibration absorbing system using compressed gas. Your first assignment requires the analysis of a non-ideal gas under pressure subject to various temperature conditions. Your goal is to maintain the pressure between 1.1 and 1.2 atm by regulating the temperature. You decide to apply your programming skills to write a program to find the values of pressure and temperature.

From your notes, you find the equation for non-ideal gases, written below, where *P* is the absolute pressure of the gas (atm), *V* is the molar volume of the gas (L/mol), *R* is the gas constant (0.0820574587 L*atm/(K*mol)), *T* is the absolute temperature (K), *a* (L^2 * atm / mol^2), and *b* (L/mol) are constants:

$$P = \frac{RT}{V - b} - \frac{a}{V^2}$$

Begin by creating a flow diagram (include it in your previously created PDF file). Then, write a Python program to solve the task. V, a, and b are given and can be assumed to be constant. An initial value of T is a user input. The functions to calculate the pressure and the new temperature (if needed) <u>must</u> be saved in a separate Python module. If the pressure is out of range, the program should find the smallest magnitude increment/decrement of temperature in order to have an acceptable value of the pressure. Your program should print on screen the initial conditions (V, a, b and T) and the resulting pressure. If the pressure is outside the range, your program should print the following information: the value of the required increment/decrement of temperature to bring the pressure into range, the resulting new temperature, and the corresponding new pressure. Present your answers in a professional way.

From your experience in ENGR14100, you are careful to make sure your code checks for errors for situations that are impossible (out of range values).

Example: Assuming the following values: $V = 18\ L/mol$, $T = 300\ K$, $a = 6.49\ L^2 \cdot atm\ /\ mol^2$, and $b = 0.0562\ L/mol$.

The outputs of your flowchart and your Python program should give the following messages:

```
Input Initial Temperature in Kelvin: 300.0
Initial conditions:
Volume = 18 L/mol
Initial temperature = 300 K
Parameter a = 6.49 L^2 atm / mol^2
Parameter b = 0.0562 L / mol
```

```
Resulting pressure = 1.35 atm
Required temperature increment for in-range pressure = -33.2 K
New temperature = 266.8 K
New pressure = 1.2 atm
```

Task 2 Files:

1) Py2_PA_Task2_main_*login*.py
2) Py2_PA_Task2_functions_*login*.py
3) Py2_PA_*login*.pdf