Lab #4 Algorithm design

1. Executable file:

import gasPriceConvertion.py file

-From {file name} import {file name};

-Get keyboard (user) input in my utility file:

-Define the input with the variable name gas\_type

-gas\_type = input("What kind of gas are you pumping? examples: regular for (R), mid-grade for (MG), or premium for (P) ? switch to the next line (\n) ");

-finish printing my output in a complete sentence, concatenate the input, use the print function and print out “your total gas price is” + gas\_type);

1. Utility file: Using elif functions to return values of the type of gas pumped.

Process gas input and output, convert gas type in letters to prices

gas\_type = float(gas\_type);

If (gas\_type != “E” or gas\_type !=“MG” or gas\_type != “P”):

Return (“something like, please enter a valid gas type”);

elif(gas\_type == upper case R or lower case r ):

-assign value to the variable regular gas price

Regular\_price = 3.7 ish (update to three digits or so);

-Define gas\_amount as user input result on how much gas user is pumping

#If the user input is a number: (if gas\_amount.isdigit()):

Nest a for loop inside to test so.

Assign a new variable total\_price and define it’s value by multiplying price of regular grade gas and the amount of gas they are pumping

total\_price = regular\_price \* gas\_amount;

Return {the pricing by multiplying the regular price with the amount of gas the user has pumped.}

If the user is not a number: (else):

Return (“not calculable, please try again”);

elif(gas\_type = “MG” or “mg” or “Mg”):

Assign how much mid-grade gas cost

midgrade\_price = 3.9 ish (update to three digits or so);

#If the user input is a number: (if gas\_amount.isdigit()):

Nest a for loop inside to test so.

(Assign a new variable total\_price and define it’s value by multiplying price of mid-grade gas and the amount of gas they are pumping)

total\_price = midgrade\_price \* gas\_amount;

Return {the pricing by multiplying the regular price with the amount of gas the user has pumped.}

If the user is not a number: (else):

Return (“not calculable, please try again”);

elif(gas\_type = “P” or “p”):

#If the user input is a number: (if gas\_amount.isdigit()):

premium\_price = 4.1 ish (update to three digits or so);

Nest a for loop inside to test so.

(Assign a new variable total\_price and define it’s value by multiplying price of premium gas and the amount of gas they are pumping)

total\_price = premium\_price \* gas\_amount;

Return {the pricing by multiplying the regular price with the amount of gas the user has pumped.}

If the user is not a number: (else):

Return (“not calculable, please try again”);

1. How to ensure first input is valid: The first if statement ensures that if user input any letters beside the ones representing the three options, r for regular, mg for mid-grade, or p for premium, no matter spacing or capitalization, they will receive an error message, otherwise, their program will carry on to the elif statements which will allow them to proceed with the second question.
2. How to ensure the second input is valid:The float conversion as well as the if else statement nested inside my if-elif statement allows me to check for the second user input to make sure the user enters a number for the question “how much gas will you be pumping? Enter a number please”, if they do not enter a number, the terminal will return some kind of error message