Test Question / if-else

TODO

A Few Questions

While loop.

Question: What is a good example of using a while loop where a for loop would make it much more difficult.

```

# Example of why to use a "while" - reverse a string

vIn = "abcd"
vOut = ""
i = len(vIn)
while ( i > 0 ):
    i = i - 1
    vOut = vOut + vIn[i]

print ( "vOut = ->{}<-".format(vOut ) )

```

or with a different way of offsetting to 0

```


# Example of why to use a "while" - reverse a string

vIn = "abcd"
vOut = ""
i = len(vIn)-1
while ( i >= 0 ):
    vOut = vOut + vIn[i]
    i = i - 1

print ( "vOut = ->{}<-".format(vOut ) )

```

Formatting.

Question: Is the `{}` a dictionary in the format statement.

```
>>> a = 1.2345
>>> print ( "X Decimal Places {}".format(a) )
>>> print ( "2 Decimal Places {:.2f}".format(a) )
>>> print ( "In Order {} second {} third {}".format( "1st", 2, "last" ) )
```

A better if-else example

Personal Income Tax Calculator. This is not all of taxes. This is just in the case where you have a job and you get a paycheck. Let’s say you have $88,000.00 a year in pay. What do the “tax” calculations mean.

First there is a standard deduction. For 2021 this is:

Amount	Description
$12,550	single taxpayers.
$12,550	married taxpayers filing separately.
$18,800	heads of households.
$25,100	married taxpayers filing jointly.

This looks like a table we can turn into an “if”/“else” in python.

```

print ( "1 for single taxpayers. " )
print ( "2 for married taxpayers filing separately.  " )
print ( "3 for heads of households.  " )
print ( "4 for married taxpayers filing jointly. " )

maritul_status = input()

standard_decuction = 12550
if maritul_status == "1" :
    standard_decuction = 12550
elif maritul_status == "2" :
    standard_decuction = 12550
elif maritul_status == "3" :
    standard_decuction = 18800
elif maritul_status == "4" :
    standard_decuction = 25100
else:
    print ( "invalid input, should be 1, 2, 3, or 4" )

print ( "Standard Deduction = {}".format( standard_decuction ) )
```

The standard deduction is take off of your income before you calculate your taxes. So the $88,000.00 minus $25,100 is: $62900.

This is the amount we use in the 2nd tax calculation.

If you search for “tax tables 2021” you get:

Tax Rate	Taxable Income Bracket	Tax Owed
10%	$0 to $14,200	10% of taxable income
12%	$14,201 to $54,200	$1,420 plus 12% of the amount over $14,200
22%	$54,201 to $86,350	$6,220 plus 22% of the amount over $54,200
24%	$86,351 to $164,900	$13,293 plus 24% of the amount over $86,350

What this table means is that you pay 10% on the first $14,200. Then take that off then pay 12% on the next chunk of money.

Let’s implement that.

```

print ( "What is your per year income" )
income_str = input()
income = float(income_str)

print ( "1 for single taxpayers. " )
print ( "2 for married taxpayers filing separately.  " )
print ( "3 for heads of households.  " )
print ( "4 for married taxpayers filing jointly. " )

maritul_status = input()


#| Amount  | Description                          |
#|---------|--------------------------------------|
#| $12,550 | single taxpayers.                    |
#| $12,550 | married taxpayers filing separately. |
#| $18,800 | heads of households.                 |
#| $25,100 | married taxpayers filing jointly.    |

standard_decuction = 12550
if maritul_status == "1" :
    standard_decuction = 12550
elif maritul_status == "2" :
    standard_decuction = 12550
elif maritul_status == "3" :
    standard_decuction = 18800
elif maritul_status == "4" :
    standard_decuction = 25100
else:
    print ( "invalid input, should be 1, 2, 3, or 4" )

print ( "Standard Deduction = {}".format( standard_decuction ) )

income = income - standard_decuction

if income < 0:
    print ( "Taxable Income = 0" )
else:
    print ( "Taxable Income = {}".format ( income ) )



#| Tax Rate | Taxable Income Bracket | Tax Owed
#|----------|------------------------|--------------------------------------------
#| 10%      | $0 to $14,200          | 10% of taxable income
#| 12%      |  $14,201 to $54,200    | $1,420 plus 12% of the amount over $14,200
#| 22%      |  $54,201 to $86,350    | $6,220 plus 22% of the amount over $54,200
#| 24%      |  $86,351 to $164,900   | $13,293 plus 24% of the amount over $86,350

tax = 0
if income >= 0 and income <= 14200:
    tax = (10/100) * income

if income >= 14201 and income <= 54200:
    tax = (10/100) * 14200
    tmp = income - 14200

    tax = tax + (12/100) * tmp

if income >= 54201 and income <= 86350:
    tax = (10/100) * 14200
    tax = tax + (12/100) * ( 54200 - 14200 )

    tmp = income - 54201
    tax = tax + (22/100) * tmp

if income >= 86350 and income <= 164900:
    tax = (10/100) * 14200
    tax = tax + (12/100) * ( 54200 - 14200 )
    tax = tax + (22/100) * ( 54200 - 14200 )

    tmp = income - 86350
    tax = tax + (24/100) * tmp

print ( "total tax for the year = {:.2f}".format(tax) )
print ( "total tax for the year = {:.2f}".format(tax) )

print ( "What is monthly witholding" )
withold_str = input()
withold = float(withold_str)

owe = tax - ( 12 * withold )
if owe < 0 :
    print ( "You get a tax refund of {:.2f}".format(-owe) )
elif owe == 0:
    print ( "You don't owe any and you don't get a refund" )
elif owe > 0:
    print ( "Send the IRS: {:.2f}".format(owe) )


```

Import

Most of the time when you build a program you have multiple files. Python deals with this with the “import” statement.

The 2 most commonly used formats are:

```
import file
```

and

```
from file imort function
```

Let’s try it (this is the all in one directory version):

```

import jane

jane.janefunc()

```

```

def janefunc():
    print ( "jane" )
```

```

```

Multiple Directories Version:

```

imort x.bob

x.bob.bobfunc()

```

```

def bobfunc():
    print ( "bob" )

```

or we can just import a single function

```

from x.bob import bobfunc

bobfunc()

```

Copyright