Assignment 4

Download the zipped folder "Assignment 4.zip", save to your computer, and then extract the files (right click and "Extract All"). There are starter files for Assignment 4.1 and 4.2 and grader files for each part.

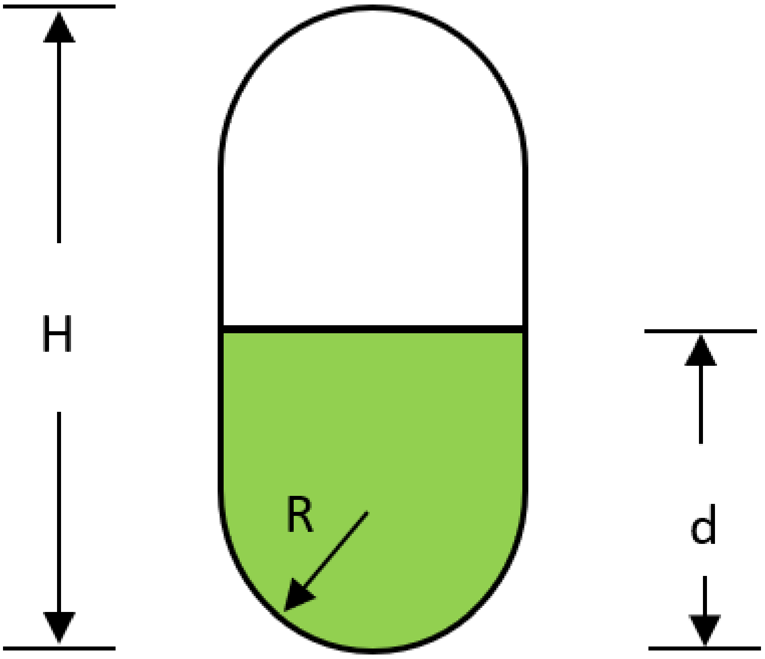
Assignment 4.zip

IMPORTANT: All subroutines (macros) and functions must be named exactly as indicated in the problem statement(s). Otherwise, the grading file will not recognize your procedures. For each assignment, I have placed one or more starter files that contain the Sub/End Sub or Function/End Function with the correct procedure names.

Assignment 4 is divided into 2 problems; each problem has a starter file available.

4.1

Create a VBA function called **tank(R,H,d)** with arguments for tank radius (R), overall tank height (H), and liquid depth (d) and the function will output the total volume (V) in the tank. (For this problem, we are assuming that the height is more than twice the radius.)



Here are the various equations that should be used:

$$\begin{array}{ll}
V = \dfrac{\pi d^2}{3}(3R - d) & d \le R \\
V = \dfrac{2\pi R^3}{3} + \pi R^2(d - R) & R < d \le H - R \\
V = \dfrac{4\pi R^3}{3} + \pi R^2(H - 2R) - \dfrac{\pi(H-d)^2}{3}(3R - H + d) & H - R < d \le H
\end{array}$$

4.2

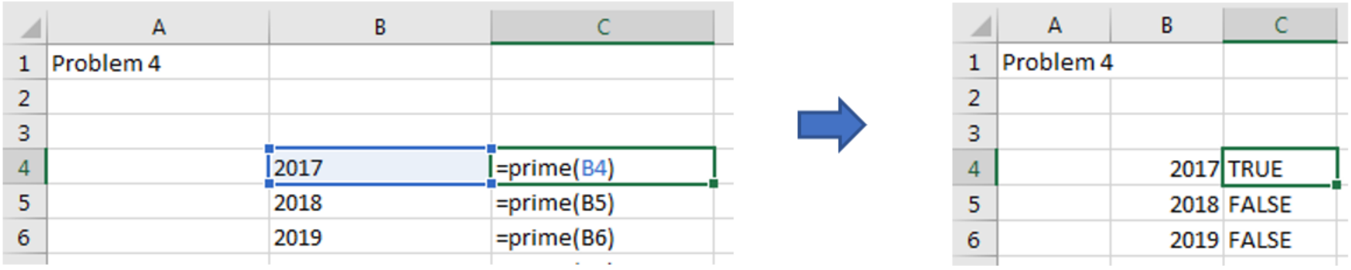
Part A: Create a VBA function called **prime(n)** that outputs TRUE if n is a prime number and FALSE if n is not a prime number. The flowchart that follows depicts the algorithm for determining if n is prime or not.

HINT #1: The **mod** function in VBA will help you determine whether a number x is divisible by y (if x mod y is equal to zero, that means x is divisible by y with no remainder).

HINT #2: The **ROUNDDOWN** function in Excel will provide you with the integer part (written in the flow chart as int) of a number. For example, int(√37) is 6 and in VBA you can use **ROUNDDOWN(SQR(37,0))**, which is 6. You could also use the **int** function.

HINT #3 (important for part B): *Make sure you treat the number 2 as a prime number!!*

Example:



	A	B	C
1	Problem 4		
2			
3			
4		2017	=prime(B4)
5		2018	=prime(B5)
6		2019	=prime(B6)

	A	B	C
1	Problem 4		
2			
3			
4		2017	TRUE
5		2018	FALSE
6		2019	FALSE

If you are curious about why we only go up through the integer just below the square root of n, you can watch the following video: https://www.youtube.com/watch?v=7VPA-HjjUmU

Part B: Create a VBA function called **countprime(n1,n2)** that counts the number of prime numbers between the integers **n1** and **n2**(inclusive). If done prudently, you should use your **prime** function from part (a).

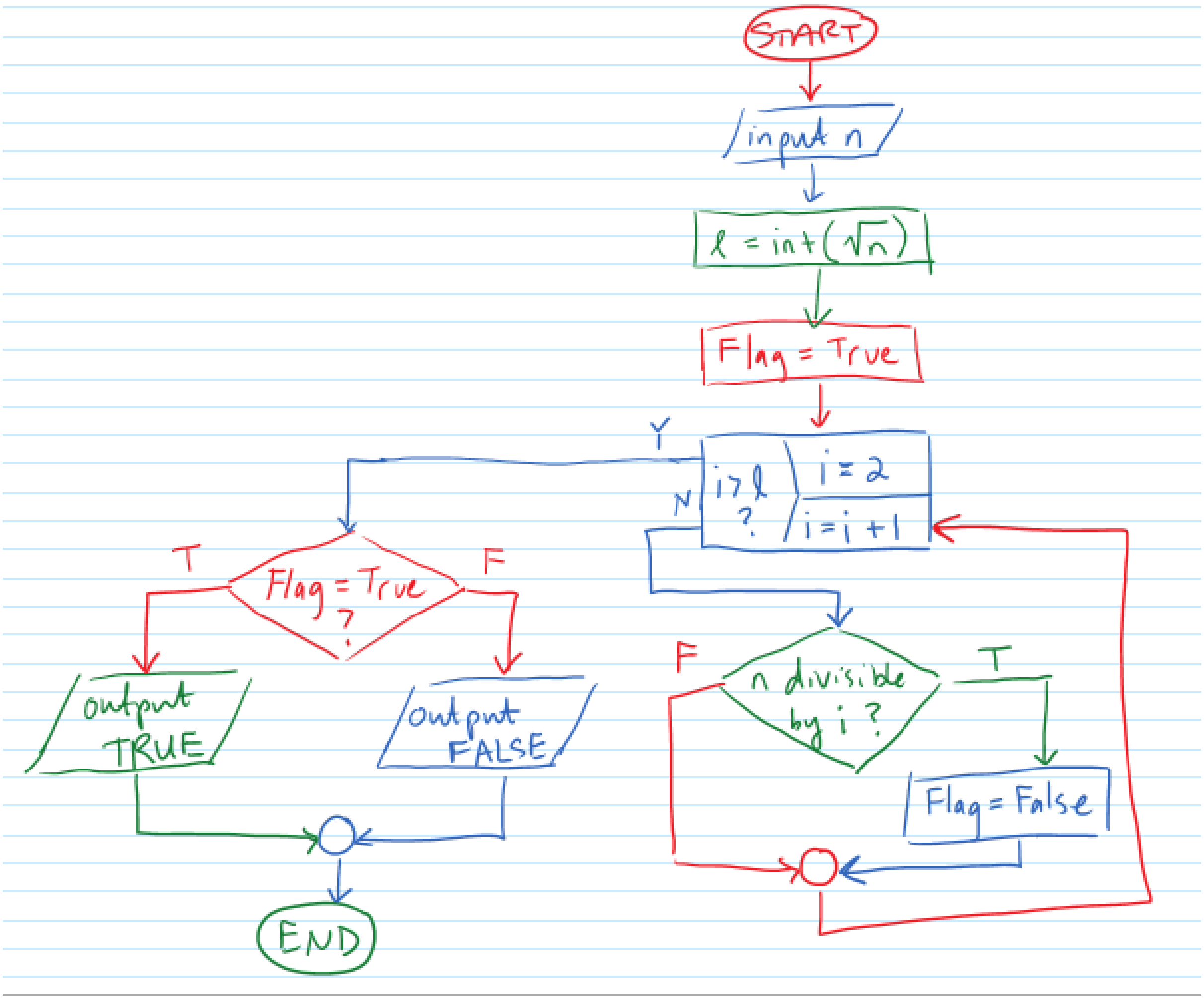
When you feel that your procedure(s) are working properly, open up the “Assignment 4 – GRADER.xlsm” file, which will check your work. If correct, you will be provided a completion code, which you can enter into the Coursera website in the "Assignment 4 submission" quiz.

Flowchart for determining if n is prime:

HINT #1: The variable **Flag** should be Dim'med as a **Boolean** (can only take on True or False).

HINT #2: Don't over-think this part. If your prime function from part A works properly, then you can just implement a simple for loop in your code and a counter variable:

```
...
For i = n1 to n2
    If prime(i) Then c = c + 1
Next i
...
```


START
input n
ℓ = int($\sqrt{n}$)
Flag = True
Y
N
i > ℓ ?
i = 2
i = i + 1
T
Flag = True ?
F
output TRUE
output FALSE
END
F
n divisible by i ?
T
Flag = False