Practice Problems:

These aren't exactly like questions that will appear on the exam, but they're Python programs on a similar scale. Write out your solution on paper, or notepad on your computer then type them into IDLE to make sure they work.

When grading problems on an exam, we care more about your logic than we do about your syntax. If you forget a colon after a conditional, no big deal. If you forget that modulo gives you a remainder, that's a problem.

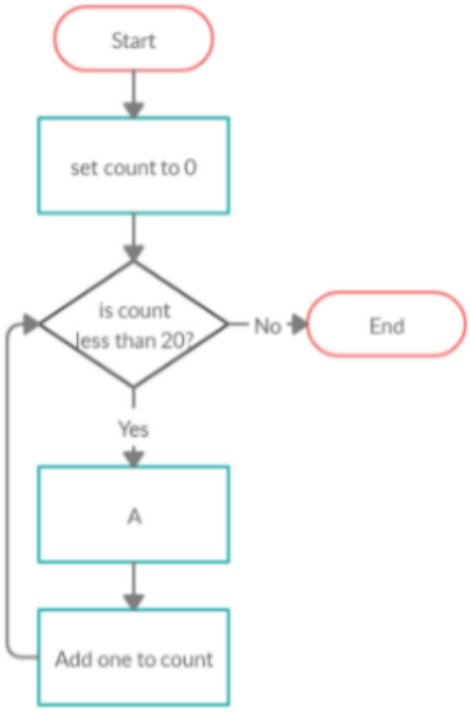
Practice Question #1:

How many times will "Hello World" be printed in the following code:

```
i= 0
while i < 10:
      if (i % 2 == 0):
            print("Hello World")
            i += 2
      else:
            i += 1
```

Practice Question #2:

How many times does A execute in the following flowchart?



Practice Question #3:

Python has a zip() function that interleaves elements of a sequence. For this problem, you'll write your own version of it.

Write a function that has 2 parameters: both are lists of strings. You can assume that the two lists are the same length. Your function should return a list that contains (at each index) the combined string from the original strings at the indices in the original list. You must use loops to solve this problem; no Python list(or zip) functions, other than len and append.

Your function must have the following specs:

- Name: **interleave**
- Input: Two lists of strings
- Returns: One list (same length as the original lists) containing the combined strings

For example, given the inputs below, your program should return the corresponding values.
Input to the Function Returns

Input to the Function	**Returns**
`['e', 'g', 'b', 'd', 'f'], ['very', 'ood', 'oy', 'oes', 'ine']`	`['every', 'good', 'boy', 'does', 'fine']`
`['Kiki ', 'you ', 'me'], ['do', 'love', '? (signed: Drake)']`	`['Kiki do', 'you love', 'me? (signed: Drake)']`
`[] , []`	`[]`

Additional thought question (at least "think" about this, even if you don't do it): How would you take the elements in the interleaved list to print a single message string like the following:

```
['every', 'good', 'boy', 'does', 'fine']
every good boy does fine

['Kiki do', 'you love', 'me? (signed: Drake)']
Kiki do you love me? (signed: Drake)
```

Practice Question #4:

What do the following expressions evaluate to? What is the data type of each one? (Write your answers FIRST, and then verify by typing them into interactive mode.)

Expression	Value	Data Type of the Value?
`20 % 7`		
`90 / 45`		
`3 * 8.0`		
`'Hey' != 'hey'`		

Practice Question #5:

Write a Python function that finds and returns the maximum element in a list. Your function must meet the following specifications:

- Name: **get_max**

- Parameter: a list of integers
- returns : an integer, the largest element in the list

The list might contain negative numbers, and it might contain duplicates. If the list is empty, return 0. Yes, we know Python supplies a **max()** but this is to practice your computational thinking. So, you must use loops to solve this problem, and **no Python list functions** are permitted, **other than len().**

For example, given the inputs below, your program should return the corresponding values. Input to the Function Returns

Input to the Function	**Returns**
`[1, 4, 9, -19]`	`9`
`[]`	`0`
`[-1, -18, -6, -250, -1]`	`-1`

Practice Question #6:

What would the following piece of code print to the screen?

```
lst = [[1, 4], [2, 5], [3, 6]]
for i in range(len(lst[0])):
      for j in range(len(lst)):
            print(lst[j][i])
```

Practice Question #7:

Write a Python function that meets the following specifications:

- Name: **flip**
- Parameters: list of booleans
- Returns: nothing
- Does: modifies the list such that every True becomes False and every False becomes True

For example, given the inputs below, your program should modify the list to hold the corresponding values. Remember, sometimes we want to **mutate** our list data that is passed in, and sometimes we don't. In this case, **the function you're writing IS mutating the list.**

Input to the Function	Returns
`[True, False, True]`	`[False, True, False]`
`[True]`	`[False]`
`[]`	`[]`

Practice Question #8: Email Processing **hard** (more than 1 solution)

Considering a typical Northeastern Email (lastname.firstname@northeastern.edu) as a string input, write a function that takes the string and returns a nested list in the form [[lastname, firstname], [northeastern, edu]].

So for example the input “kim.sam@northeastern” into this function should return [[‘kim’, ‘sam’], [‘northeastern’, ‘edu’]]. (Hint: use the .split() function)

```
def email_process(email):
```

Practice Question 9:

Taking the output of #8 (a list of lists) as input into the following function, write a function that recreates the email.
So for example the input [[‘kim’, ‘sam’], [‘northeastern’, ‘edu’]] into this function should return “kim.sam@northeastern.edu”

```
def recreate_email(list):
```

Sample Solutions For Practice Problems

Note: Be sure to TRY THE PROBLEM on your own first, before looking at these sample solutions. Also note that(as always) there may be more than ONE solution - these samples may not be the only way to solve the problem!

#1: How many times will “Hello World” be executed?

5 times

#2: How many times will A be executed?

20 times

#3:Interleave

```
def interleave(a, b):
     new_list = []
     for i in range(len(a)):
          new_list.append(a[i] + b[i])
     return new_list
```

#4: What do they evaluate to?

Expression	Value	Data Type of the Value?
`20 % 7`	`6`	`Integer`
`90 / 45`	`2.0`	`Float`
`3 * 8.0`	`24.0`	`Float`
`‘Hey’ != ‘hey’`	`False`	`Boolean`

#5 Get max

```
def get_max(a_list):
     if not a_list:
          return 0

     current_max = a_list[0]
```

```
    for each in a_list:
        if each > current_max:
            current_max = each
    return current_max
```

#6: What would the following piece of code print to the screen?

1
2
3
4
5
6

#7: Flip

```
# Solution A
def flip(lst):
    for i in range(len(lst)):
        lst[i] = not(lst[i])

# Solution B
def flip(lst):
    for i in range(len(lst)):
        if lst[i] == True:
            lst[i] = False
        else:
            lst[i] = True
```

#8: Email Processing **h

ard** (more than 1 solution)

```
def email_process(email):
    result = []
    split = email.split("@")
    result.append(split[0].split("."))
    result.append(split[1].split("."))
    return result
```

#9: Recreate Email

```
def recreate_email(list):
      name = list[0][0] + "." + list[0][1]
      domain = "@" + list[1][0] + "." + list[1][1]

      return name + domain
```