Name: _____ andrewID: _____

Instructions:

- Do not open this exam until the instructor tells you to.
- This exam tests material from weeks 1-4 of the course. You have **50 minutes**.
- If you have a clarification question, raise your hand and a proctor will come help you.
- You must complete the exam **individually**. You may refer to paper notes during the exam, but do not communicate with anyone else.
- Show your work to get full credit.
- Make sure to display values **unambiguously**. Strings should be in quotes (unless they're being printed), floats should have decimals, Booleans should be capitalized.

#1 - Code Writing - Function Definitions [14pts]

Write the function `semester(h1, h2)`. This function takes two integers as parameters, representing study hours for two weeks. It then generates a random number of hours between 5 and 40 (inclusive) for a third week. The function prints the third week's hours and it returns the average of the three weekly hour counts rounded to the nearest integer.

For example, if we call `semester(18, 28)` and the randomly generated hour is 22, the function prints "`Week 3: 22`" and returns 23 (average of 18, 22, 28 rounded to the nearest integer.)

Hint: recall that the `random` library has a function `randint(a, b)` that generates a random number in the inclusive range [a, b]. There is also a built-in `round(x)` function.

Note: if you want to use a library, you must import it.

#2 - Code Reading - Conditionals [13pts]

Consider the following piece of code:

```
def fn(p, q, r):
    if p % 2 == 1:
        if q < 20:
            print("A")
        elif q < 50:
            print("B")
        if r > 3:
            print("C")
    elif q < 10:
        if p >= 2 and q < 15:
            print("D")
            if p < 25 or q > 5:
                print("E")
    else:
        if r == 5:
            print("F")
        else:
            print("G")
        print("H")
```

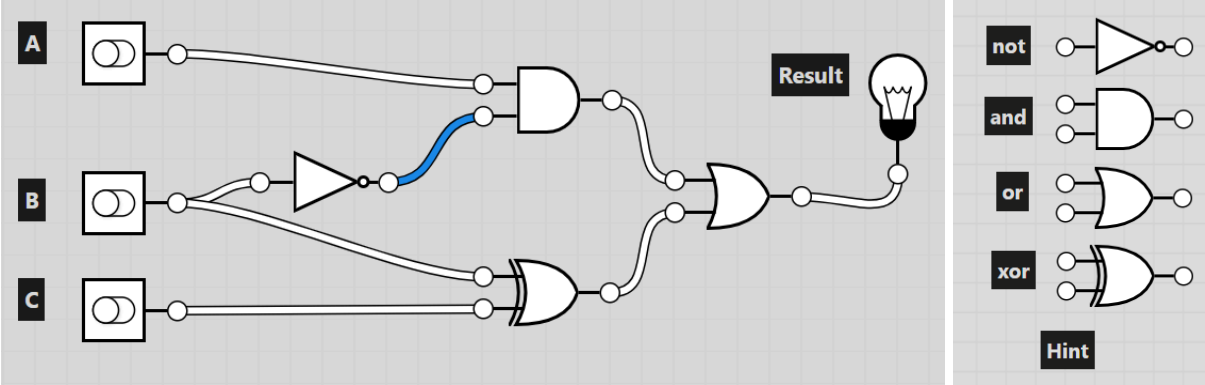
For each row, write in the cell what will be printed if we run the code above it.

`fn(6, 12, 5)`	`fn(11, 25, 5)`	`fn(4, 2, 5)`

Is there an input that would make `fn` print both **A** and **B**? If yes, give an example. If no, explain why.

#3 - Free Response - Circuits and Gates [15pts]

Translate the following circuit to a Boolean expression, and fill out a truth table that shows all possible results of the circuit. The table may have more space than you need.



Note: we haven't included a header row for the truth table - make sure to write it in. You can include additional columns to show your work if you want them.

Truth table:

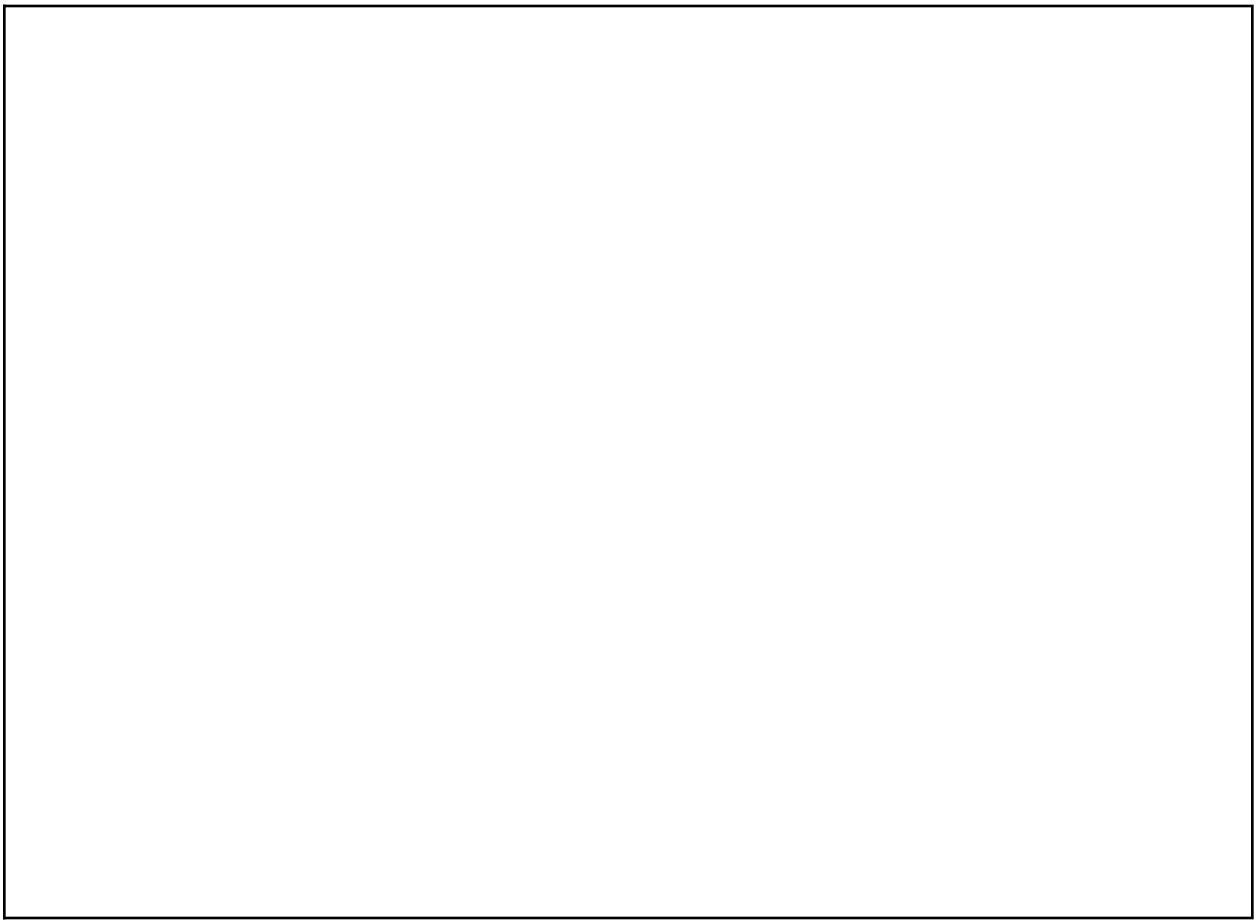
Boolean expression:

#4 - Code Writing - Loops [15pts]

Write the function `countVowelPairs(s)` which takes `s`, a string, and returns the number of adjacent vowel pairs that occur in s. Consider a, e, i, o, u as vowels (all lower case - no need to handle uppercase vowels).

```
countVowelPairs("beautiful") == 2  # "ea" and "au"
countVowelPairs("queue") == 3      # "ue", "eu", "ue"
countVowelPairs("hello") == 0      # no adjacent vowels
countVowelPairs("aeiou") == 4      # "ae", "ei", "io", "ou"
countVowelPairs("a") == 0          # no pairs
```

Note: when more than two vowels appear next to each other, **every** pair of letters counts as a vowel pair. So `"aei"` counts as two pairs, `"aeio"` counts as three, etc.



#5 - Code Reading - Nesting [14pts]

Consider the following code:

```
def f(x, y):
    count = 0
    for i in range(x):
        if x < y:
            for j in range(i, y):
                print("in:", i, j)
        else:
            count = count + 1
    return count

print("pre")
print("f:", f(2, 3))
print("post")
```

What will be printed when we run this code?

Is it possible to call this function on a set of arguments such that it returns the value 4? If yes, provide an example function call that does so. If no, explain why not.

#6 - Short Answer - Data Types [4pts]

For each of the following expressions, write the eventual **value** of the expression and that value's **type**.

Expression	Value	Type
`2 ** 4`		
`"12" * 3`		
`7.5 - 4`		
`(2 + 3) == 7`		

#7 - Short Answer - Error Messages [4pts]

For each of the following snippets of code, select whether it causes a Syntax Error, Runtime Error, or neither (choose only one answer each). **Assume that no variables are defined before the code runs.**

`print(2 * 3 == 12)` ☐ Syntax Error ☐ Runtime Error ☐ Neither

`pow(x, 3)` ☐ Syntax Error ☐ Runtime Error ☐ Neither

```
for i in range(3):
print(i)
```
☐ Syntax Error ☐ Runtime Error ☐ Neither

```
x = 5
y - 3 = x
```
☐ Syntax Error ☐ Runtime Error ☐ Neither

#8 - Short Answer - Loop Control Variables [6pts]

You want to print all the **odd numbers** between 3 and 99, inclusive of both (so 3, 5, 7, 9…). This could be implemented using a while loop. Write snippets of code to implement the **start value**, **continuing condition**, and **update action** for the loop control variable you would use in that while loop.

In the loop's body the loop control variable will be printed before the update action happens, and no if statement will be used.

Start Value	Continuing Condition	Update Action

#9 - Short Answer - Scope [3pts]

Consider the following code snippet:

```
def convert(score):
    fraction = score / 100
    return round(fraction, 2)

x = 92.35
print("Your score:", convert(x))
```

List all the **local variables** in this code snippet.

List all the **global variables** in this code snippet.

#10 - Short Answer - Binary [6pts]

For each of the following problems, you must show your work to receive full credit.

A: Convert **00100111** from binary to decimal.

Work	
Answer	

B: Convert **26** from decimal to binary.

Work	
Answer	

#11 - Short Answer - Indexing and Slicing [6pts]

Given the string

```
s = "supercalifragil"
```

what would each of the following expressions evaluate to?

`s[len(s)-3]`	
`s[5:]`	
`s[2:12:4]`	