CSc 110 Sample Midterm Exam #1

1. Expressions

For each expression in the left-hand column, indicate its value in the right-hand column.
Be sure to list a constant of appropriate type (e.g., `7.0` rather than `7` for a `float`, `String`s in quotes).

Expression	Value
`8 + 5 * 3 / 2`	_____
`1.5 * 4 * 7 // 8 + 3.4`	_____
`73 % 10 - 6 % 10 + 28 % 3`	_____
`4 + 1 + 9 + (-3 + 10) + 11 // 3`	_____
`3 // 14 // 7 / (1.0 * 2) + 10 // 6`	_____
`10 > 11 == 4 / 3 > 1`	_____
`not(2 >= 11 or 10 < 67 or 4 / 4 >= 1)`	_____
`(True or not 2 < 3) and 6 == 4 / 3`	_____

2. Parameter Mystery

At the bottom of the page, write the output produced by the following program.

```
def main():
    x = "happy"
    y = "pumpkin"
    z = "orange"
    pumpkin = "sleepy"
    orange = "vampire"

    orange(y, x, z)
    orange(x, z, y)
    orange(pumpkin, z, "y")
    z = "green"
    orange("x", "pumpkin", z)
    orange(y, z, orange)

def orange(z, y, x):
    print(y, "and", z, "were", x)
```

3. If/Else Simulation

For each call of the function below, write the value that is returned:

```
def mystery(n):
    if n < 0:
        n = n * 3
        return n
    else:
        n = n + 3
    if n % 2 == 1:
        n = n + n % 10
    return n
```

Function Call	Value Returned
`mystery(-5)`	_____
`mystery(0)`	_____
`mystery(7)`	_____
`mystery(18)`	_____
`mystery(49)`	_____

4. Programming

Write a function `print_rectangles` that takes two integers as parameters, a width and a height. Your function should prompt the user for a scale factor (an integer). The function prints two rectangles consisting of ASCII stars ("*"). The first rectangle is of size width times height. The second rectangle is shifted to the right by the width of the first rectangle and is width * scale wide and height * scale high.

Function call	`print_rectangles(5, 3)`	`print_rectangles(4, 2)`

	`Scale = 2`
	
	`*****`
	`*****`
	`*****`
	`          **********`
	`          **********`
	`          **********`
	`          **********`
	`          **********`
	`          **********`
		`Scale = 3`
		
		`****`
		`****`
		`          ************`
		`          ************`
		`          ************`
		`          ************`
		`          ************`
		`          ************`

5. Programming

Write a function named `print_grid` that accepts two integer parameters *rows* and *cols*. The output is a grid of numbers where the first parameter (*rows*) represents the number of rows of the grid and the second parameter (*cols*) represents the number of columns. The numbers count up from 1 to (*rows* x *cols*). The output are displayed in column-major order, meaning that the numbers shown increase sequentially down each column and wrap to the top of the next column to the right once the bottom of the current column is reached.

Assume that *rows* and *cols* are greater than 0. Here are some example calls to your function and their expected results:

Call	`print_grid(3, 6)`	`print_grid(5, 3)`	`print_grid(4, 1)`	`print_grid(1, 3)`

Output	`1 4 7 10 13 16`
	`2 5 8 11 14 17`
	`3 6 9 12 15 18`
		`1 6 11`
		`2 7 12`
		`3 8 13`
		`4 9 14`
		`5 10 15`
			`1`
			`2`
			`3`
			`4`
				`1 2 3`

6. Programming

Write a function called `hopscotch` that shows a sequence of numbers similar to the appearance of a hopscotch board. (Hopscotch is a "hopping" game played by children.) For this problem, a hopscotch board is an increasing sequence of numbers in lines in an alternating pattern. Odd lines contain a single number indented by 3 spaces. Even lines contain a pair of numbers with the first number not indented, followed by 5 spaces, followed by the second number.

Your function should accept a parameter representing the number of "hops" worth of numbers to display. A board of 0 "hops" shows only the number 1, indented by 3 spaces. A "hop" is defined as a trio of additional numbers spread across 2 additional lines, following the pattern described previously. So for example, 1 hop means to show the numbers 1 then 2-3-4 to the board in the pattern shown below. 2 hops means to show the numbers 1 then 2-3-4, 5-6-7 on the board. 3 hops means to show 1 then 2-3-4, 5-6-7, 8-9-10, and so on. Assume that the number of hops passed will be at least 0.

Here are some example calls to the function and their resulting console output:

Call	`hopscotch(0)`	`hopscotch(1)`	`hopscotch(2)`	`hopscotch(3)`	`hopscotch(5)`

output	`  1`	`  1`
		`2  3`
		`  4`
			`  1`
			`2  3`
			`  4`
			`5  6`
			`  7`
				`  1`
				`2  3`
				`  4`
				`5  6`
				`  7`
				`8  9`
				`  10`
					`  1`
					`2  3`
					`  4`
					`5  6`
					`  7`
					`8  9`
					`  10`
					`11  12`
					`  13`
					`14  15`
					`  16`

CSc 110 Sample Midterm Exam #1 Key

1. Expressions

Expression	Value
`8 + 5 * 3 / 2`	`15.5`
`1.5 * 4 * 7 // 8 + 3.4`	`8.4`
`73 % 10 - 6 % 10 + 28 % 3`	`-2`
`4 + 1 + 9 + (-3 + 10) + 11 // 3`	`24`
`3 // 14 // 7 / (1.0 * 2) + 10 // 6`	`1.0`
`10 > 11 == 4 / 3 > 1`	`False`
`not (2 >= 11 or 10 < 67 or 4 / 4 >= 1)`	`False`
`(True or not 2 < 3) and  6 == 4 / 3`	`False`

2. Parameter Mystery

```
happy and pumpkin were orange
orange and happy were pumpkin
orange and sleepy were y
pumpkin and x were green
green and pumpkin were vampire
```

3. If/Else Simulation

Function Call	Value Returned
`mystery(-5)`	`-15`
`mystery(0)`	`6`
`mystery(7)`	`10`
`mystery(18)`	`22`
`mystery(49)`	`52`

4. Programming (four solutions shown)

```
def print_rectangles(w, h):
    scale = int(input("Scale = "))
    for i in range(h):
        print("*" * w)
    for i in range(scale * h):
        print(" " * w, end='')
        print("*" * (w * scale))
```

5. Programming (one solutions shown)

```
def print_grid(rows, cols):
    for i in range(1, rows + 1):
        for j in range(cols):
            print(i + rows * j, "", end='')
        print()
```

6. Programming