CSc 110 Sample Midterm Exam #2

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`, `str`s in quotes).

Expression	Value
`1 + 2 * 3 - 4 * 5`	_____
`5 // 2 + 9.0 / 2.0 - 2 * 1.25`	_____
`29 % 2 % 5 + 34 % 3`	_____
`8 + 6 * -2 + 4 + (2 + 5) > 5`	_____
`31 // 2 / 10.0 + 10 / (5 / 2.0)`	_____
`(1 != 2) != (2 != 3)`	_____

2. Parameter Mystery

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

```
def main():
    a = "felt"
    b = "saw"
    c = "drew"
    saw = "sue"
    drew = "b"

    mystery(a, b, c)
    mystery(b, a, saw)
    mystery(drew, c, saw)
    mystery("a", saw, drew)
    mystery(a, a, "drew")

def mystery(b, a, c):
    print(c, a, "the", b)

main()
```

3. If/Else Simulation

For each call of the function below, write the output that is produced:

```
def mystery(x, y):
    if x > y:
        x = x - 5
        y = y + 5
    if x < y:
        x += 1
        y -= 1
    else:
        x = y * 2
    print(x, y)
```

Function Call	Output
`mystery(4, 7)`	_____
`mystery(3, 3)`	_____
`mystery(10, 5)`	_____
`mystery(20, 4)`	_____
`mystery(1, 1)`	_____

4. Programming

Write a function called `rand_square` that takes an integer `size` as a parameter and prints a `size` by `size` square where each line has a series of 0 or more "\" characters followed by 0 or more "/" characters. Your function should use `random.randint` to choose the number of "\"s to print on each line (a number between 0 and `size` inclusive). The function should return the total number of characters in the square.

The following table lists some calls to your function and one of their possible expected outputs. Keep in mind that since the number of "\" characters on each line is random, your output may not match exactly.

Call	`rand_square(5)`	`rand_square(3)`	`rand_square(1)`	`rand_square(0)`	`rand_square(10)`
Example Output	\\\\/ \\\\\ \\/// \\/// \\\//	\// \// \\/	\		\\\\\\\\// ////////// \\\\\\\\// \\\\\\//// \\\\\\\\// \\\\\\\\// \\\\\\\\// \\\\\\//// \\\\\\\\//
Return Value	25	9	1	0	100

5. Programming

Write a function `print_range` that takes three integers as parameters, a low and a high and a number of rows to print. It should print all numbers in range low to high inclusive on the first line and print similar ranges on the rest of the lines. Each line should start one number higher. Numbers should be aligned with the same number on the above line. You can assume that numbers passed in will be positive.

Function call	**Output**
`print_range(5, 10, 2)`	`5 6 7 8 9 10` `  6 7 8 9 10`
`print_range(3, 3, 1)`	`3`
`print_range(10, 8, 87)`	

`print_range(1, 4, 3)`	`1 2 3 4`
	`  2 3 4`
	`  3 4`

8. Programming

Write a function named **`random_rects`** that asks a user how many rectangles they want and then prompts them for a width and height for each rectangle. It then outputs all of the rectangles made of ascii stars and their combined area. The user will be guaranteed to input valid positive integers for each value when prompted.

The following calls demonstrate your function's behavior. Bold and underlined text is user input. Your function should match this output format exactly:

Call

```
random_rects()
```

Output

```
How many rectangles? 3
Width 1? 2
Height 1? 3
**
**
**
Width 2? 3
Height 2? 2
***
***
Width 3? 10
Height 3? 1
**********
Total area: 22
```

Call

```
random_rects()
```

Output

```
How many rectangles? 4
Width 1? 5
Height 1? 2
*****
*****
Width 2? 4
Height 2? 2
****
****
Width 3? 3
Height 3? 2
***
***
Width 4? 2
Height 4? 2
**
**
Total area: 28
```

CSc 110 Sample Midterm Exam #2 Key

1. Expressions

Expression	Value
`1 + 2 * 3 - 4 * 5`	`-13`
`5 // 2 + 9.0 / 2.0 - 2 * 1.25`	`4.0`
`29 % 2 % 5 + 34 % 3`	`2`
`8 + 6 * -2 + 4 + (2 + 5) > 5`	`True`
`31 // 2 / 10.0 + 10 / (5 / 2.0)`	`5.5`
`(1 != 2) != (2 != 3)`	`False`

2. Parameter Mystery

```
drew saw the felt
sue felt the saw
sue drew the b
b sue the a
drew felt the felt
```

3. If/Else Simulation

Method Call	Output
`mystery(4, 7)`	`5 6`
`mystery(3, 3)`	`6 3`
`mystery(10, 5)`	`6 9`
`mystery(20, 4)`	`18 9`
`mystery(1, 1)`	`2 1`

4. Programming (five solutions shown)

```
def rand_square(size):
    count = 0
    for i in range(size):
        backslashes = random.randint(0, size)
        count += backslashes
        print("\\" * backslashes, end="")
        print("/" * (size - backslashes))
    return count
```

5. Programming (three solutions shown)

```
def print_range(low, high, rows):
    start = low
    for i in range(rows):
        print("  " * i, end="")
        for j in range(start, high + 1):
            print(str(j) + " ", end="")
        start += 1
        print()
```

6. Programming

```
def random_rects():
    area = 0
    count = int(input("How many rectangles? "))
    for i in range(1, count + 1):
        width = int(input("Width " + str(i) + " ? "))
        height = int(input("Height " + str(i) + " ? "))

        for j in range(1, height + 1):
            for k in range(1, width + 1):
                print("*", end='')
            print()

        area += (width * height)

    print("Total area: " + str(area))
```