Lecture 3 - Exercises

3A - Operators and Operator Precedence

1. Evaluate each of the following expressions. Be sure to properly indicate the type of the value, for example by including a decimal point if the result is a `float` or enclosing it in double quotes if it's a `str`.
    1. `9 / 3`
    2. `9 // 3`
    3. `10 // 3`
    4. `10 % 3`
    5. `3 % 10`
    6. `2 ** 3`
    7. `"abc" + "def"`
    8. `"baa " * 2`
2. Evaluate each of the following expressions. As before, be sure to be precise about the type of the resulting value.
    1. `(9 % ( 6 // 2))`
    2. `9 % 6 // 2`
    3. `2 ** 2 ** 4`
    4. `("na" * 8 + " ") * 2 + "Batman!"`
    5. `1 + 2 ** 3 / 4 * 5 - (6 % 7)`

3B - Program Execution and Return Values

3. For each of the following, say whether it is a statement or an expression:

    1. `a = 4`
    2. `a + 4`
    3. `int(6.4) + 2`
4. Suppose we run the following program, and the user types `6` and presses enter. What happens?

```
user_num = input("Enter a number: ")
result = 5 % (3 ** (user_num // 4))
```

5. Suppose we run the following program, and the user types `6` and presses enter. What value gets stored in `result`?

```
user_num = int(input("Enter a number: "))
result = 5 % (3 ** (user_num // 4))
```

6. What does the following expression evaluate to? `float(str(int(2.6 / 2))*2)`

Problems

1. Write a program that prompts the user for three numbers in a row, then prints the sum of all three numbers. The program should work if the user enters decimal numbers, but can throw an error if the user enters something that isn't a number. An example run of the program might look like this, where the numbers after each prompt are typed by the user:

```
Enter the first number: 4
Enter the second number: 2.4
Enter the third number: 8.6
Your numbers total 15.0
```

2. Write a program to help a user calculate percentages. First prompt the user for an amount, then prompt the user for a percentage. Finally, print the percentage of the amount. The amounts and percents should handle floating-point values, but your program may throw an error if the user enters something that's not a number. A sample run of the program might look like this:

```
Enter an amount: 900
Enter a percentage: 50.0
50.0 percent of 900 is 450.0
```

3. Modify your program from #1 to print the full equation for the sum; for example, the last line of the example above would read

   ```
   4.0+2.4+8.6=15.0
   ```

   The catch: you have to print the equation without spaces between the numbers.