Lecture 7 - Exercises

7A - While Loops

1. What does the following program print?

    ```
    count = 10
    while count < 21:
        print(count, end=" ")
        count += 3
    ```

2. Consider the following program:

    ```
    n = 12345
    m = 0
    while n != 0:
        m = (10 * m) + (n % 10)
        n //= 10
    ```

    1. How many times is the body of the `while` loop executed?
    2. What are the values of `m` and `n` after the code is executed?

3. Consider the following code, which has two missing snippets, marked `[[1]]` and `[[2]]`.

    ```
    i = [[1]]
    while [[2]]:
      i += 1
      print(i)
    ```

    Consider the following candidates to replace the missing snippet `[[1]]`:

    a. `0`

    b. `1`

    c. `10`

    and the following candidates to replace missing snipped `[[2]]`:

    i. `i < 9`

    ii. `i <= 9`

    iii. `i < 10`

    iv. `i <= 10`

    To form a complete program, we need to pick a pair of snippets, one to replace `[[1]]` and one to replace `[[2]]`.

    1. Which pair or pairs of snippets will cause the program to print the numbers from 1 to 10, inclusive?
    2. Which pair or pairs of snippets will cause the program to print exactly 10 numbers?

7B - Nested while loops

4. Consider the following program:

```
i = 0
while i < 4:
  j = 1
  while j < 11:
    print('*', end='')
    j += 1
  i += 1
  print()
```

1. How many asterisks (`*`) are printed by the program?
2. What are the values of `i` and `j` at the end of the program?

Problems

1. Write a program that prompts the user for a positive integer p and prints the powers of 2 up to 2^p.

```
Enter a power: 4
2^0: 1
2^1: 2
2^2: 4
2^3: 8
2^4: 16
```

2. Write a program that prompts the user for a positive integer n and prints the powers of 2 less than n.

```
Enter a number: 9
2^0: 1
2^1: 2
2^2: 4
2^3: 8
```

3. Write a program that prompts the user until they enter the a secret password correctly. For this problem, the secret password can be whatever you want and should be hard-coded into the program. For example, if I chose `"banana"` as the secret password, a run of the program might look like this:

```
Enter the password: algebra
Incorrect, try again: bookend
Incorrect, try again: banana
You're in!
```

4. Write a program that repeatedly prompts a user for positive numbers until a negative number is entered, then print the sum of the positive numbers entered (be sure not to include the negative number in the sum).

```
Enter a number: 4
Enter a number: 8
Enter a number: 3
Enter a number: -9
The positive numbers entered sum to 15
```

5. Write a program that prints a multiplication table for all possible combinations of the numbers 1 through 6. Ideally, print spaces as needed to pad out single-digit numbers so the table’s columns line up neatly.

```
1  2  3  4  5  6
2  4  6  8 10 12
3  6  9 12 15 18
4  8 12 16 20 24
5 10 15 20 25 30
6 12 18 24 30 36
```

6. (Challenge Problem!) Write a program that prompts the user for a positive integer and prints the binary representation of that integer, with no leading zeros.

```
Enter a positive integer: 65
65 in binary is 1000001
```