Topics

- Getting user input
- Manipulating strings (a few ways)

Question 1:

Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.

Extras:

1. Add on to the previous program by asking the user for another number and printing out that many copies of the previous message. (*Hint: order of operations exists in Python*)
2. Print out that many copies of the previous message on separate lines. (*Hint: the string* `"\n` *is the same as pressing the ENTER button*)

Question 2:

Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. *Hint: how does an even / odd number react differently when divided by 2?*

Extras:

1. If the number is a multiple of 4, print out a different message.
2. Ask the user for two numbers: one number to check (call it `num`) and one number to divide by (`check`). If `check` divides evenly into `num`, tell that to the user. If not, print a different appropriate message.

Question 3:

Take a list, say for example this one:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

and write a program that prints out all the elements of the list that are less than 5.

Extras:

1. Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.
2. Write this in one line of Python.
3. Ask the user for a number and return a list that contains only elements from the original list `a` that are smaller than that number given by the user.

Question 4:

Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don't know what a *divisor* is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)

Question 5:

Take two lists, say for example these two:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

Extras:

1. Randomly generate two lists to test this
2. Write this in one line of Python (don't worry if you can't figure this out at this point - we'll get to it soon)

Question 6:

Ask the user for a string and print out whether this string is a palindrome or not. (A **palindrome** is a string that reads the same forwards and backwards.)