Section 2: Welcome to Python

Moonweight

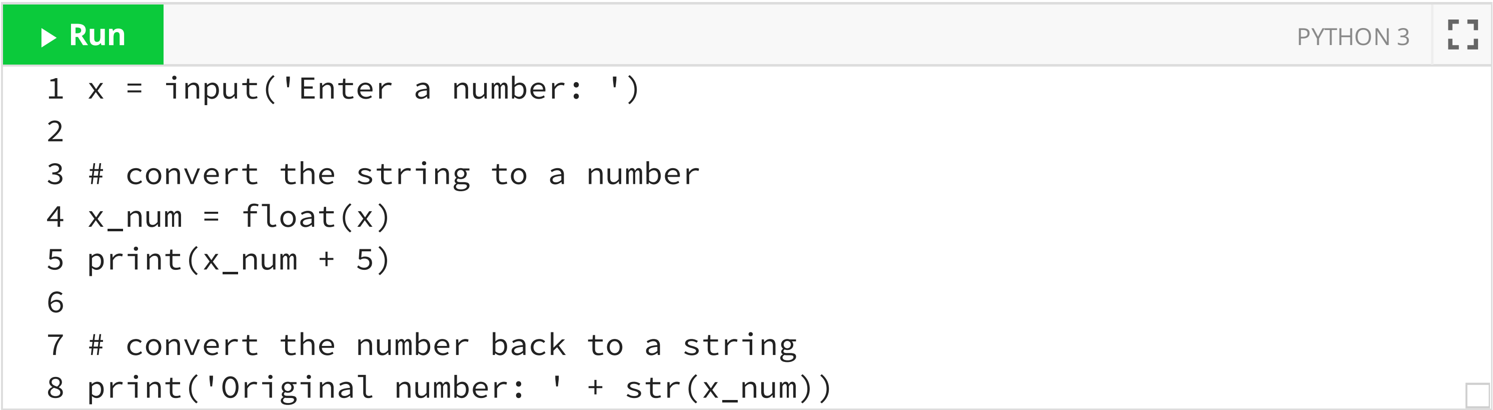
Write a Python program that calculates a user's weight on the moon, given their weight on earth.

The user (an earthling) enters a weight on earth. Your program will print the corresponding weight on the moon, which is 16.5% of the weight on earth.

Here's an example of the program running:

```
$ python moonweight.py
Enter a weight on earth: 120
Equivalent weight on moon: 19.8
```

Tip - be sure to convert strings to floats in Python before performing operations:



```python
x = input('Enter a number: ')

# convert the string to a number
x_num = float(x)
print(x_num + 5)

# convert the number back to a string
print('Original number: ' + str(x_num))
```

8-ball: Milestone 1

The idea behind a magic 8-ball is very simple. You ask the eight ball a yes or no question, and it tells you the answer. Except, that the answer it chooses is randomly selected from a set of prefabricated responses.

Here's what a physical magic 8-ball looks like:

Write a program that continuously prompts the user for a yes or no question, and then randomly selects from a set of canned answers: The classic 8-ball responses were:

```
As I see it, yes.
Ask again later.
Better not tell you now.
Cannot predict now.
Concentrate and ask again.
Don’t count on it.
It is certain.
It is decidedly so.
Most likely.
My reply is no.
My sources say no.
Outlook not so good.
Outlook good.
Reply hazy, try again.
Signs point to yes.
Very doubtful.
Without a doubt.
Yes.
Yes definitely.
You may rely on it.
```

You do not have to have so many responses. Instead choose at least 5, or make up your own.

Here's some demos of the program running:

```
$ python eightball.py
Ask a yes or no question: Is Karel married?
Not a chance.
```

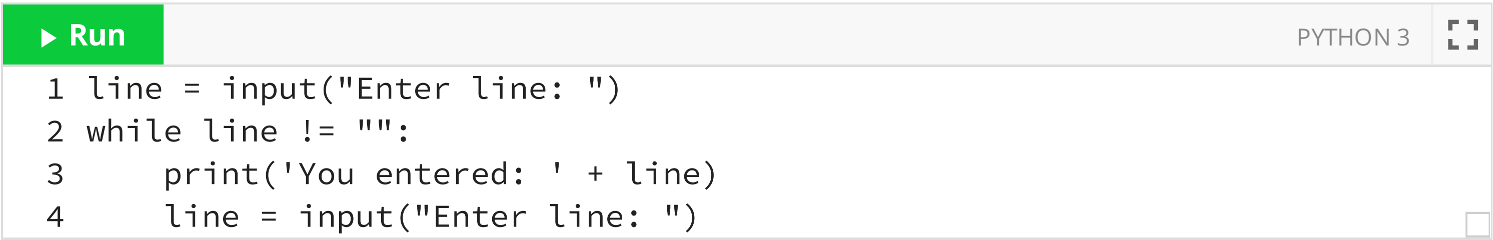
```
$ python eightball.py
Ask a yes or no question: Is my real name Chris?
Only Karel knows.
```

```
$ python eightball.py
Ask a yes or no question: 8-ball, are you using random numbers?
Without a doubt.
```

> i Note: As this exercise is open ended, there's no automatic feedback. Though be sure to run and test your own code!

8-ball: Milestone 2 (optional)

If you have time, try to extend your 8-ball solution so that the user can repeatedly answer questions until the user enters no question.



```python
line = input("Enter line: ")
while line != "":
    print('You entered: ' + line)
    line = input("Enter line: ")
```