Hints - Oblig 1

We recommend that you test your functions with simple examples first.

softmax_loss_naive

To make the first softmax task easier, we provide some simple input with correct solutions, so you have a 100% accurate solution.

You can generate the data like this:

```
np.random.seed(1)
W = np.random.randn(2, 3)
X = np.random.randn(3, 2)
y = np.arange(3)
```

This should give the following values:

W:
[[1.62434536 -0.61175641 -0.52817175]
 [-1.07296862 0.86540763 -2.3015387]]

X:
[[1.74481176 -0.7612069]
 [0.3190391 -0.24937038]
 [1.46210794 -2.06014071]]

y:
[0, 1, 2]

When you run with those numbers:
loss, grad = **softmax_loss_naive**(W, X, y, 0.0)

you should get **loss**:
0.997810

and **grad**:
[[0.33479965 -0.0874167 -0.24738295]
 [-0.47226825 0.06906057 0.40320768]]