WEEKLY PARTICIPATION 10

In the implementation of an RNN-based classifier used in Lecture 23, we defined the forward pass of the model as follows[1]:

```
def forward(self, x):
        xembed = self.embeddings(x)
        h0 = torch.zeros(self.num_layers, xembed.size(1),
                         self.hidden_size).to(device)
        hiddenStates, _ = self.RNN(xembed, h0)
        out = hiddenStates[-1, :, :]
        return self.logits(out)
```

Questions:

(1) If `x` is a $T \times B$ tensor of indices into the vocabulary, the word embedding dimension is D, the number of layers is 1, and the hidden state size for the RNN is H, what are the dimensions of `xembed`, `h0`, `hiddenStates`, `out` and `self.logits(out)`?

(2) Read the documentation for `torch.nn.RNN`, and give a replacement for the last three lines of this `forward` that uses the *second* output argument of `torch.nn.RNN` instead of the first.

[1] The source listing has been edited, but makes the same computations.