Question 1:

a) 32, if the **gpus** memory was full because the model parameters, so when use multi **gpus** each **gpu** will copy the model on their memory so the remaining space will hold one instance.
b) In pytorch **DISTRIBUTED DATA PARALLEL (DDP**) is responsible for parallelism at small scale for each component in the model to parallelize and DISTRIBUTED the model.
c) At first, I will check cuda toolkit for both the machine and environment, and gpus drivers if each exist and with version combatable to my setup, then I will check my frame work installation.

Question 2

There several reasons for exact same code produce different results every time:

1. **If not Setting a random seed for the code**
2. **Using could change it usages and location there as different hardware can lead to result change.**
3. **Different libraries versions.**

Please Note: I got issue with python 3.6 environment pytorch doesn't recognize the gpu so I created author environment with python 3.8, but. All my notebooks runs on both environments but I preferred the one with gpu enabled.

Also the notebook ''Spectral-Normalization implementation and testing '' contain how Implemented and tested Spectral Normalization code.

I'm eager to receive your feedback.