PROJECT 1

In this assignment you are tasked with writing a node that subscribes to a topic and publishes to another. Your code will subscribe to a topic called '`two_ints`', on which a custom message containing two integers can be broadcast. Make sure to familiarize yourself with the message format of this topic (have a look at the `TwoInts.msg` in the `msg` directory). Those two integers are to be added and the result published to topic '`sum`' as an `Int16` from `std_msgs`.

Setup

- Access your Vocareum workspace for Project 1
- Start by running `source setup_project1.sh` in the command line terminal. You should do this first every time you load or reload your workspace. You must run this command before trying to invoke any ROS commands (`catkin_make`, `roscd`, etc.). This will also start a `roscore` for your session.
- Once you have sourced this script, there will be a ROS package publishing random integers to the '`two_ints`' topic every two seconds.

- **Please do not start your own `roscore`.**
- **Please do not use `./setup_project1.sh` to source your workspace, this is not supported by Vocareum.**

Implementation

- You must implement your code in the file `~/catkin_ws/src/project1_solution/scripts/solution.py` . This file has already been created for you and any starter code has been placed inside.

Testing

To test your code, you have multiple options:

1. Add some debug output to your publisher (i.e. print the two numbers you have just received as well as their sum to the console every time you are about to publish). Then simply run your node (`rosrun project1_solution solution.py`). This is useful to see that you are getting to the right place in your code, but will not tell if you are actually publishing, and publishing to the right topic.

2. Simply submit your code. Your code will be automatically graded and after a short while you should find a 'Submission Report' under the 'Details' tab which will contain output from the grading script.

3. (Requires more advanced Linux skills) Run your node in the background (`rosrun project1_solution solution.py &`). This frees up the console so you can manually subscribe to the `sum` topic (`rostopic echo sum`) and see that something is indeed being published. Of course, you'll then need to manually kill your node, by retrieving the right process number and sending it a `SIGINT` signal using the `kill` command.