1 Pintos-IC Task 0 questions

1. Which Git command should you run to retrieve a copy of your group's shared Pintos repository in your local directory?
2. Why is using the `strcpy()` function to copy strings usually a bad idea?
3. Explain how thread scheduling in Pintos currently works in less than 250 words. Include the chain of execution of function calls.
4. Explain the property of reproducibility and how the lack of reproducibility will affect debugging.
5. How would you print an unsigned 64 bit `int`? (Consider that you are working with C99)
6. What makes locks and semaphores in Pintos similar? What extra property do locks have that semaphores do not?
7. What are the limitations on the size of the thread struct? How does Pintos identify stack overflow?
8. If test '`src/tests/threads/alarm-multiple`' fails, where would you find its output and result logs? (Hint: you might want to run this test to find out.)
9. Given a struct defined as follows:

   ```
   struct foo
   {
     int bar;
     struct list_elem e;
   };
   ```

   And a list declaration:

   ```
   struct list foo_list;
   ```

   Give a piece of code that would insert an element of `struct foo` into the list ordered (in ascending order) by the element `bar`. You may assume that '`<list.h>`' has been included. (Hint: you will probably find it useful to reread you C notes from last year.)
10. For a list of `struct foo` as defined above, write a piece of code to iterate through the list and return a pointer to the struct if the element `bar` is equal to some `int x`.