Last name _____

First name _____

LARSON—MATH 556–SAGE WORKSHEET 08
Four Invariants: α, α', β, and β'

1. Log in to your Sage/Cocalc account.

   (a) Start Chrome browser.

   (b) Go to `http://cocalc.com` and "Sign In".

   (c) Click project **Classroom Worksheets**.

   (d) Click "New", call it **s08**, then click "Sage Worksheet".

   We'll start by defining some built-in graphs in Sage and then use Sage to test if the graphs are bipartite and to calculate the four invariants: α, α', β, and β'.

2. Evaluate `pete = graphs.PetersenGraph()`. Now evaluate `pete.order()` and `pete.size()`. To see what the graph looks like, evaluate `pete.show()`.

3. Is the Petersen graph bipartite? Sage has a built-in test to check. Evaluate `pete.is_bipartite()`. What do you get?

4. Let `k_3_4 = graphs.CompleteBipartiteGraph(3,4)`. Use Sage to find the order and size of $K_{3,4}$. Use Sage to draw the graph.

5. Is $K_{3,4}$ bipartite? Use Sage to check.

   Now let's compute some invariants. While we can write our own algorithms, Sage often has the state-of-the-art (fastest) algorithms built-in.

6. To find a maximum independent set in the Petersen graph, evaluate `pete.independent_set()`. What do you get?

   To find out the options for this function, evaluate `pete.independent_set?`. One of the options shows you how to return the independence number α.

7. According to the help you read, we can use `pete.independent_set(value_only=True)`. Evaluate.

   If we wanted an independence number function we could use this. Evaluate the following code:

```
def independence_number(g):
    return g.independent_set(value_only=True)
```

8. Now try `independence_number(pete)`. What did you get?

9. What can you type to find the independence number of $K_{3,4}$?

The vertex covering number β is not a built-in Sage function. But using the Gallai identities, and our independence number function, we can write our own function to calculate β:

```
def vertex_covering_number(g):
    return g.order() - independence_number(g)
```

10. Now try `vertex_covering_number(pete)`. What did you get?

11. What can you type to find the vertex covering number of $K_{3,4}$?

12. To find a maximum matching in the Petersen graph, evaluate `pete.matching()`. What do you get?

To find out the options for this function, evaluate `pete.matching?`. One of the options shows you how to return the matching number α'.

If we wanted a matching number function we could define the following function. Evaluate.

```
def matching_number(g):
    return g.matching(value_only=True)
```

13. Now try `matching_number(pete)`. What did you get?

14. What can you type to find the matching number of $K_{3,4}$?

The edge covering number β' is not a built-in Sage function. But using the Gallai identities, and our matching number function we can write our own function to calculate ρ:

```
def edge_covering_number(g):
    return g.order() - matching_number(g)
```

15. Now try `edge_covering_number(pete)`. What did you get?

16. What can you type to find the edge covering number of $K_{3,4}$?

17. We weren't careful about one thing with our edge covering number function. What