Last name _____

First name _____

LARSON—MATH 305—CLASSROOM WORKSHEET 03
Sage LAB!—Section 1.1

1. Log in to your Sage/CoCalc account.

   (a) Start the Chrome browser.

   (b) Go to `http://cocalc.com` and sign in.

   (c) You should see an existing Project for our class. Click on that.

   (d) Click "New", call it **c03**, then click "Sage Worksheet".

   (e) For each problem number, label it in the Sage cell where the work is. So for Problem 1, the first line of the cell should be `#Problem 1`.

   (f) When you are finished with the worksheet, click "make pdf", email me the pdf (at `clarson@vcu.edu`, with a header that says **Math 305 c03 worksheet attached**).

The following examples are all from our text. We'll just see today that we can compute them in Sage/Cocalc.

2. `prime_range(10,50)`.

3. `[n for n in range(10,30) if not is_prime(n)]`.

4. `gcd(97,100)`.

5. `gcd(97  10^15, 19^20  97^2)`.

6. `factor(1275)`.

7. `factor(2007)`.

8. `factor(31415926535898)`.

9.
```
n = 740375634795617128280467960974295731425931888 8
...923128908493623263897276503402826627689199641962511 7
...843995894330502127585370118968098286733173273108930 9
...005525051168770632990723963807867100860969625379346 5
...0563796359
```

10. `len(n.str(2))`.

11. `len(n.str(10))`.

12. `n.is_prime()` .

13. `p = 2^32582657 - 1`.

14. `p.ndigits()`.

15. Find the first few Euclidean primes. Let $P_1 = 2$. Then at each step find the product of the existing primes plus 1. Add the largest prime factor that is not in your current list of Euclidean primes.

16. (**Density of the Primes**). Find the ratio of the number of primes in the interval $[10^i]$ to 10^i for $i = 1 \ldots 9$.

Boolean Expressions in Sage

A *boolean expression* is one that evaluates to True or False.

17. Evaluate `3==4`.

18. Evaluate `3==3`.

19. Evaluate `3>3`.

20. Evaluate `3>=-3`.

21. Evaluate `13%2==1`.

22. Evaluate `13%2==0`.

While "==" is used as a claim of equality of expressions (the left-hand-side and the right-hand-sides of the "==") the symbol "!=" is used to express does-not-equal.

23. Evaluate `5!=7`.

24. Evaluate `5!=5`.

25. We will *assign* a value to a variable "a". Then we will use that variable in a boolean expression. (These two lines can be typed in one cell, or each in its own cell). Type and evaluate:

```
a=5
a>2
```

Boolean expressions can be combined with *boolean operators* like "and" and "or".

26. Evaluate `3==3 and 3==4`.

27. Evaluate `3==3 or 3==4`.

Lists in Sage

A *list* is a basic *data structure* in Python and Sage. They are represented by square brackets with comma separated numbers, strings, etc., between them (like $[2, 5, 9]$ or `["red", "blue"]`).

28. Lists can be given names. Evaluate `L=[2,5,9]`. Then evaluate `L`.

29. Lists are indexed starting with 0. Evaluate each of `L[0]`, `L[1]`, `L[2]`, and `L[3]`.

30. Lists can be combined with "+". Evaluate `[2,5,9]+[3,4,5]`.

31. Let `M=[3,4,5]`. Evaluate `L+M`.

32. If you want all the integers from x to y you can use the shorthand notation $[x..y]$. Evaluate `[3..7]`.

33. If you want a list with m n's you can use the shorthand notation $[n] * m$. Evaluate `[0]*7`.

34. You can have a list of lists. Evaluate `L=[[0,1],[2,3],[4,5]]`. Now evaluate `L[1]`. Then evaluate `L[1][0]`. What do you think the value of `L[0][1]` is?