1. Log in to your Sage Cloud account.
   
   (a) Start the Chrome browser.
   (b) Go to http://cloud.sagemath.com and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click “New”, call it c10, then click “Sage Worksheet”.

Warm-up

2. Consider the system:

\[
\begin{align*}
9a + 3b + 1c &= 32 \\
4a + 2b + 1c &= 15 \\
a + b + c &= 6
\end{align*}
\]

Find a matrix that represents this system, find the row-reduced echelon form of this matrix, rewrite this as an equivalent system of linear equations and interpret.

Programming

A for loop is what we use when we want our code to run through every item \(x\) in a list.

3. Evaluate and test the following function. What do you think this function will do?

```python
def for_loop_test():
    for i in [0..5]:
        print i**2
```

4. Modify your code to print the squares of the integers from 5 to 9. How did you change it?

5. Modify the code to print just the squares of 2, 5, 7, 9, and 23. How did you change it?
6. The function `list_evens(n)` that returns all the even integers from 0 to \( n \). Evaluate and test the following code.

```python
def list_evens(n):
    M=[]
    for x in [0..n]:
        if x%2==0:
            M.append(x)
    return M
```

7. Write a function `list_primes(n)` that returns a list of all the primes up to \( n \). Test it.

8. Write a function `count_primes(n)` that returns a count of all the primes up to \( n \). Test it.

A *while loop* runs a block of code while a condition is still satisfied.

9. Type in and evaluate the function `while_test()`. What do you think this function will do?

```python
def while_test():
    i=0
    while i<5:
        print i^2
        i=i+1
```

10. Now modify your function to produce a similar function that prints the squares of the integers from 5 to 9.

11. A common way to use a while loop is in a test where you don’t know precisely when the test condition will be met. Here we will write a function that finds which letter of a word is the first occurrence of the letter “a”. The program prints “no a’s when there is no “a” in the word.

```python
def find_first_a(word):
    length=len(word)
    i=0
    while i<length:
        if word[i]="a":
            return i
        else:
            i=i+1
    print "{} contains no a’s".format(word)