1. Log in to your Sage/Cocalc account.
   (a) Start the Chrome browser.
   (b) Go to \texttt{http://cocalc.com} and sign in.
   (c) You should see an existing Project for our class. Click on that.
   (d) Click “New”, call it \texttt{c08}, 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 \texttt{#Problem 1}.

2. \textbf{Warm-ups}. What will the following commands return in Sage? Answer and then
use Sage to check.
   (a) \texttt{is\_prime(245); is\_prime(23)}
   (b) \texttt{even = lambda x: x\%2==0; even(6); even(7)}
   (c) \texttt{L=[2..10]; len(L)}
   (d) \texttt{[k**2 for k in L]}  
   (e) \texttt{n=6}
   \hspace{1cm} \texttt{if n<7:}
   \hspace{2cm} \texttt{print "{} is less than 7".format(n)}
   
   (f) \texttt{n=6}
   \hspace{1cm} \texttt{if n<5:}
   \hspace{2cm} \texttt{print "{} is less than 7".format(n)}
   
   (g) \texttt{i=0}
   \hspace{1cm} \texttt{while i<7:}
   \hspace{2cm} \texttt{print i}
   \hspace{2cm} \texttt{i=i+1}

3. \textbf{Matrices Review}. Consider the system:
\[
\begin{align*}
9a + 3b + 1c &= 32 \\
4a + 2b + 1c &= 15 \\
1a + 1b + 1c &= 6
\end{align*}
\]
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.

4. \textbf{Programming}. What will the following code do?
\begin{verbatim}
def mystery1(n):
    L=[1..n]
    even = lambda x: x\%2==0
    print [even(k) for k in L]
mystery1(9)
\end{verbatim}
A while loop runs a block of code while a condition is still satisfied. 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.

5. Type in and evaluate the function while_test().

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

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

7. 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)
```

8. What will the following code do?

```python
def mystery3(n):
    M=[]
    i=0
    even = lambda x: x%2==0
    while i<n:
        if even(i)==True:
            M.append(i)
        i=i+1
    print M
mystery3(9)
```

9. Write a definition for a function that prints the lists [1..i] for i=0 to i=4. Use a while loop. Evaluate and test. Then try to write a definition for a function that prints the lists [1..i] for i=0 to i=n.