

Last name _____

First name _____

LARSON—OPER 635—SAGE WORKSHEET 03
Programming—Control Flow.

1. Log in to your Sage Cloud account.
 - (a) Start Firefox or Chrome browser.
 - (b) Go to `http://cloud.sagemath.com`
 - (c) Click “Sign In”.
 - (d) Click project **OPER 635**.
 - (e) Click “New”, call it **s03**, then click “Sage Worksheet”.

It is often useful to manipulate and/or create tuples.

2. Here is a function that takes 2 numbers as inputs and returns a tuple (pair) with twice numbers.

```
def tuple_test(x,y):  
    t=(2*x,2*y)  
    return t
```

Evaluate. Let `s=tuple_test(3,4)`. Evaluate.

3. Now write a function `pair_square(x,y)` that takes any numbers x and y and returns a tuple (pair) that is the squares of these numbers.

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

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

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

5. Modify your code to print the squares of the integers from 5 to 9. How did you change it?
6. Modify the code to print just the squares of 2, 5, 7, 9, and 23. How did you change it?

7. The function `list_evens(n)` that returns all the even integers from 0 to n . Evaluate and test the following code.

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

8. Write a function `list_primes(n)` that **returns a list** of all the primes up to n . Use Sage's built-in `is_prime(n)` function. 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?

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

10. 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 prints the first n primes. We will use a *counter* to keep track of how many we have so far. The first version works and the 2nd runs forever. Why?

```
def print_first_n_primes2(n):
    count = 0
    current_number = 0
    while count < n:
        if is_prime(current_number) == True:
            print current_number
            count = count + 1
            current_number = current_number + 1
```

```
def print_first_n_primes(n):
    count = 0
    current_number = 0
    while count < n:
        if is_prime(current_number) == True:
            print current_number
            count = count + 1
            current_number = current_number + 1
```