

1] Use the function **range** to calculate the sum of

- all multiples of 3 from 1 to 1000 (inclusive)
- the squares of all multiples of 3 between 100 and 200 (inclusive).

2] Write a script that generates a list of the first k prime numbers, where k is entered by the user.

```
primes = []
k = input('Enter how many primes: ')
count = 2
while len(primes) < k:
    '''decide if count is prime
    if it is, append it to the list'''
    count += 1
print primes
```

Here's some code to get you started:

3] We presented two methods for removing duplicates from a list:

- Building a new list, containing only unique elements, and
- Converting the list to another data type, the set (and then converting it back).

Write a program that compares the speeds of the two methods by running each one a large number of times, in a loop. For testing, use the list [1, 1, 1, 2, 3, 3, 4, 5, 5, 5], and repeat the loop 100,000 times. Time the loops with the **time** function from the **time** module, as shown in class.

4] Create automatically a two-dimensional list of size NxN, where N is entered by the user (use N = 10 for testing, but it should work for any N). The initial list elements should be all zeros. Print the list like this:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Now modify the list with a double loop to put ones on and under the antidiagonal, like this:

```
[0, 0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0, 1, 1]
[0, 0, 0, 0, 0, 1, 1, 1]
[0, 0, 0, 0, 1, 1, 1, 1]
[0, 0, 0, 1, 1, 1, 1, 1]
[0, 0, 1, 1, 1, 1, 1, 1]
[0, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1]
```