

```
In [32]: # Example 26: np.max(), np.min(), np.argmax(), np.argmin()
```

```
a = np.array([0,1,2,3,16,3,2,1,0])
print('a:',a)
print('max of a =',np.max(a))
print('min of a =',np.min(a))
print('index of max value of a =',np.argmax(a))
print('index of min value of a =',np.argmin(a))
```

```
a: [ 0  1  2  3 16  3  2  1  0]
max of a = 16
min of a = 0
index of max value of a = 4
index of min value of a = 0
```

## Questions

These questions are in no particular order (except for question 0, do that one first). The questions range in difficulty; some are one-liners, others require a lot more thinking. Don't be discouraged if you hit a roadblock. Talk to your neighbors and ask for help from the lab staff.

### Question 0

**In order to test your code, please run the cell below to load the autograder. There is a cell after each question that you can run in order to check your answer. The autograder is purposefully not very verbose.**

```
In [33]: %run autograder.py
```

```
Autograder loaded!
Question 0 Passed!
```

### Question 1

**Search the NumPy documentation and/or the web for a NumPy function that can solve a system of linear equations of the form  $Ax=b$ . Once you've found the package and function, insert into the `func` placeholder below. Often Googling `NumPy` followed by the function you'd like will give you exactly what you want. Ex: Try searching `NumPy Matrix Solver`**

```
In [40]: # find the missing package/function
func = np.linalg.solve # YOUR CODE HERE

# Do not modify the code below
def q1(A,b):
    return func(A,b)
```

```
In [41]: test_q1(q1)
```

```
Question 1 Passed!
```

## Question 2

**Given NumPy array A, return an array that consists of every entry of A that has an even row index and an odd column index.**

See [slicing](#) for examples on how to do array slicing.

```
In [38]: def q2(A):
          """
          Input:
          A - MxN NumPy array

          Output:
          Returns an NumPy array that consists of every entry of A that has an ev

          Example:
          A = np.array([[ 1  2  3  4  5]
                        [ 6  7  8  9 10]
                        [11 12 13 14 15]
                        [16 17 18 19 20]
                        [21 22 23 24 25]])

          Output = np.array([[ 2  4]
                             [12 14]
                             [22 24]])

          """

          # YOUR CODE HERE
          return A[::2, 1::2]
```

```
In [39]: test_q2(q2)
```

Question 2 Passed!

## Question 3

**Given an MxN NumPy array, first find the indices of the maximum value in each row of the array, then return the maximum index.**

Hint: There is a function to find the index of the maximum value in a vector.

Hint 2: List Comprehensions and Iterators might be useful.