

HW 6 - NumPy Arrays

Due March 11th, 2025 at 11:59pm

1 Prime Clusters

You have obtained a dataset of star temperatures from different stellar clusters. For your research, you are interested only in clusters where **at least one** star's temperature is a prime number. Given a 2D NumPy array, write a function to find the rows where **at least one** value is a prime number. For example:

```
>>> arr = np.array([[2, 3, 5], [4, 6, 8], [11, 13, 17], [7, 10, 13]])
>>> containsPrimes(arr)
array([[2, 3, 5],
       [11, 13, 17],
       [7, 10, 13]])
```

2 Let's play Checkers!

You've decided to take a break from your cutting-edge research and play checkers with your friend. Unfortunately, there is no checkerboard in sight! Therefore you must create one yourself.

2.1

Start by writing a function that creates a 8x8 square matrix with only zeros.

```
>>> checkerboard()
array([[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]])
```

2.2

For only the odd rows, make an alternating pattern of ones and zeroes.

```
>>> checkerboard()
array([[1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0]])
```

2.3

Finish the checkerboard with the even rows.

```
>>> checkerboard()
array([[1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1]])
```

2.4

Re-write your function such that the checkerboard begins with a 0 instead.

```
>>> reverse_checkerboard()
array([[0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 1, 0, 1, 0, 1, 0, 1],
       [1, 0, 1, 0, 1, 0, 1, 0]])
```

3 The Expanding Universe

You have now become fascinated with how dark energy is making galaxies accelerate away from us. Write a function that takes in a string and a number,

and returns the string with the specified number of spaces inserted between each letter, simulating the expansion of space! For example:

```
>>> universe = np.array(['galaxy', 'clusters'])
>>> expansion(universe, 1)
array(['g a l a x y', 'c l u s t e r s'])

>>> expansion(universe, 2)
array(['g a l a x y', 'c l u s t e r s'])
```

4 Second-Dimmest Star

While analyzing a dataset of star luminosities, you need to identify the **second-dimmest** star in each cluster. Write a function that takes a 2D NumPy array and returns an array containing only the second-smallest value in each column. For example:

```
>>> np.random.seed(123)
>>> stars = np.random.randint(500, 2000, (5, 5))
array([[1123, 1456, 1789, 1324, 1876],
       [1567, 1987, 1678, 1405, 1589],
       [1345, 1654, 1523, 1109, 1923],
       [1298, 1890, 1367, 1784, 1432],
       [1823, 1756, 1489, 1672, 1550]])
>>> secondDimmest(stars)
array([1298, 1654, 1489, 1324, 1550])
```