Extracting Elements from Array Description From a given array, extract all the elements which are greater than 'm' and less than 'n'. Note: 'm' and 'n' are integer values provided as input.

Input format:

A list of integers on line one

Integer 'm' on line two

Integer 'n' on line three

Output format:

1-D array containing integers greater than 'm' and smaller than 'n'.

Sample input:

[1, 5, 9, 12, 15, 7, 12, 9] (array)

6 (m)

12 (n)

Sample output:

[9 7 9]

```
import numpy as np
```

```
arr = np.array([ 1, 5, 9, 12, 15, 7, 12, 9 ])
```

```
m = int(input('Enter m: '))
n = int(input('Enter n: '))
```

```
     Enter m: 6
     Enter n: 12
```

```
arr[((arr > m) & (arr < n))]
```

```
     array([9, 7, 9])
```

Print "+" Description Given a single positive odd integer 'n' greater than 2, create a NumPy array of size (n x n) with all zeros and ones such that the ones make a shape like '+'. The lines of the plus must be present at the middle row and column.

Hint: Start by creating a (n x n) array with all zeroes using the np.zeros() function and then fill in the ones at the appropriate indices. Use integer division (//) to access the middle rows and columns

Examples:

Input 1:

3

Output 1:

[[0 1 0]

[1 1 1]

[0 1 0]]

Input 2:

5

Output 1:

[[0 0 1 0 0]

[0 0 1 0 0]

[1 1 1 1 1]

[0 0 1 0 0]

[0 0 1 0 0]]

Explanation: Notice that the 1s in the arrays make a shape like '+'.

```
size = int(input('Enter shape: '))
zer = np.zeros((size,size))
plus = int(size/2)
```

```
     Enter shape: 7
```

```
zer[plus,:] = 1
zer[:,plus] = 1
zer
```

```
     array([[0., 0., 0., 1., 0., 0., 0.],
            [0., 0., 0., 1., 0., 0., 0.],
            [0., 0., 0., 1., 0., 0., 0.],
            [1., 1., 1., 1., 1., 1., 1.],
            [0., 0., 0., 1., 0., 0., 0.],
            [0., 0., 0., 1., 0., 0., 0.],
            [0., 0., 0., 1., 0., 0., 0.]])
```

✓ 0s completed at 10:52 AM ● ×