

```
In [1]: import numpy as np
```

1. Index of the 100th Element

Consider an (11,12) shape array. What is the index (x,y) of the 100th element? Note: For counting the elements, go row-wise. For example, in the array:

[[1, 5, 9], [3, 0, 2]] the 5th element would be '0'.

Answer:-

(8,3)

Alternative 1 -

```
In [2]: np.unravel_index(99, (11,12)) # indexing starts from 0 so 100th element is 99
```

```
Out[2]: (8, 3)
```

Alternative 2 -

```
In [3]: arr1 = np.arange(1, (11*12)+1).reshape(11,12)
        np.argwhere(arr1==100)
```

```
Out[3]: array([[8, 3]], dtype=int64)
```

2. Reshaping an Array

Suppose you have an array 'p':

[[1, 5], [3, 7], [4, 9]] What will be the output of the following code? np.reshape(p, -1)

Answer:-

1-d array

3. Reshaping an Array

Consider the array provided below:

```
[[1, 2, 3, 4, 5] [6, 7, 8, 9, 10] [11, 12, 13, 14, 15] [16, 17, 18, 19, 20]]
```

Now, you are expected to generate the following array out of it:

```
[[1, 3] [5, 7] [9, 11] [13, 15] [17, 19]]
```

Which code will give you the correct output? (Hint: Also check if one or more options are correct.)

Answer:-

```
print(array_1[array_1%2 != 0].reshape(5, 2))
```

In [4]:

```
array_1 = np.array([[1, 2, 3, 4, 5],[6, 7, 8, 9, 10],[11, 12, 13, 14, 15],[16, 17, 18, 19, 20]])  
print(array_1[array_1%2 != 0].reshape(5, 2))
```

```
[[ 1  3]  
 [ 5  7]  
 [ 9 11]  
 [13 15]  
 [17 19]]
```

4. Reshaping an array

Suppose you have an array 'p'

[[1, 5], [3, 7], [4, 9]] What will be the output of the following code?

```
np.reshape(p, (1, -1))
```

Answer:-

2-D Array

Coding

Stacking arrays

Description - Merge the three arrays provided to you to form a one 4x4 array.

In [5]:

```
import ast
import numpy as np

input_str = input()
input_list = ast.literal_eval(input_str)
list_1 = input_list[0]
list_2 = input_list[1]
list_3 = input_list[2]

final_array = np.hstack((np.vstack((list_1,list_2)),np.transpose(list_3)))
print(final_array)
```

```
[[[7, 13, 14], [18, 10, 17], [11, 12, 19]], [16, 6, 1], [[5, 8, 4, 3]]]
[[ 7 13 14  5]
 [18 10 17  8]
 [11 12 19  4]
 [16  6  1  3]]
```
