



## Module 4: NumPy Case Study

## Problem Statement:

You work in XYZ Company as a Python developer. The company officials want you to build a Python program.

## Tasks To Be Performed:

1. Create a function that takes dimensions as tuples e.g. (3, 3) and a numeric value and returns a NumPy array of the given dimension filled with the given value e.g.: `solve((3, 3), 5)` will return

```
[
    [5, 5, 5],
    [5, 5, 5],
    [5, 5, 5]
]
```

2. Create a method that takes n NumPy arrays of the same dimensions, sums them and returns the answer.
3. Given a 2 D Array of N X M Dimension, write a function that accepts this array as well as two numbers N and M. The method should return the top-left N X M sub matrix, e.g:

```
[
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
]
```

`top_left_sub_matrix (matrix, 2, 2) -> should return:`

```
[
    [1, 2]
    [4, 5]
]
```

4. Given a 2 D Array of N X M Dimension, write a function that accepts this array as well as two numbers N and M. The method should return the bottom-right N X M sub matrix, e.g:

```
[
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
]
```

sub\_matrix(matrix, 1, 1) -> should return : (Keep in mind these arrays are zero indexed)

```
[
    [5, 6]
    [8, 9]
]
```

5. Given a 1 D NumPy Array. Write a function that accepts this array as parameters. The method should return a dictionary with 'mean' and 'std\_dev' as key and array's mean and array's standard deviation as values:

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

solution(arr) -> should return :

```
{'mean': 1.0, 'std_dev': 0.0}
```