Python Practice Questions

1) **Write a program to iterate the first 10 numbers and in each iteration, print the sum of the current and previous number.**

Expected Output:

```
Printing current and previous number sum in a range(10)
Current Number 0 Previous Number  0  Sum:  0
Current Number 1 Previous Number  0  Sum:  1
Current Number 2 Previous Number  1  Sum:  3
Current Number 3 Previous Number  2  Sum:  5
Current Number 4 Previous Number  3  Sum:  7
Current Number 5 Previous Number  4  Sum:  9
Current Number 6 Previous Number  5  Sum:  11
Current Number 7 Previous Number  6  Sum:  13
Current Number 8 Previous Number  7  Sum:  15
Current Number 9 Previous Number  8  Sum:  17
```

2) **Write a program to find how many times substring “Emma” appears in the given string.**

Given:

```
str_x = "Emma is good developer. Emma is a writer"
```

Expected Output:

```
Emma appeared 2 times
```

3) **Create a new list from a two list using the following condition**

Given a two list of numbers, write a program to create a new list such that the new list should contain odd numbers from the first list and even numbers from the second list.

Given:

```
list1 = [10, 20, 25, 30, 35]
list2 = [40, 45, 60, 75, 90]
```

Expected Output:

```
result list: [25, 35, 40, 60, 90]
```

4) **Print downward Half-Pyramid Pattern with Star (asterisk)**

```
* * * * *
* * * *
* * *
* *
*
```

5) **Write a function called `exponent(base, exp)` that returns an int value of base raises to the power of exp.**

Note:- here `exp` is a non-negative integer, and the base is an integer.

Expected output

Case 1:

```
base = 2
exponent = 5

2 raises to the power of 5: 32 i.e. (2 *2 * 2 *2 *2 = 32)
```

Case 2:

```
base = 5
exponent = 4

5 raises to the power of 4 is: 625
i.e. (5 *5 * 5 *5 = 625
```

6) Write a NumPy program to test whether none of the elements of a given array is zero.
7) . Write a NumPy program to create an array of 10 zeros,10 ones, 10 fives.
8) Write a NumPy program to create an array of all the even integers from 30 to70.
9) Write a NumPy program to compute sum of all elements, sum of each column and sum of each row of a given array.
10) Write a NumPy program to create a 5x5 zero matrix with elements on the main diagonal equal to 1, 2, 3, 4, 5.
11) Write a NumPy program to sort a given array by row and column in ascending order.
12) Write a NumPy program to create a three-dimension array with shape (3,5,4) and set to a variable.
13) Write a NumPy program to swap rows and columns of a given array in reverse order.Write a NumPy program to multiply two given arrays of same size element-by-element.
14) Write a NumPy program to print the NumPy version in your system.
15) Print max from axis 0 and min from axis 1 from the following 2-D array.

```
sampleArray =
numpy.array([[34,43,73],[82,22,12],[53,94,66]])
```

Expected Output:

```
Printing Original array


[[34 43 73]
```

```
 [82 22 12]

 [53 94 66]]

 Printing amin Of Axis 1

 [34 12 53]

 Printing amax Of Axis 0

 [82 94 73]
```

16) Delete the second column from a given array and insert the following new column in its place.

```
sampleArray =
numpy.array([[34,43,73],[82,22,12],[53,94,66]])
newColumn = numpy.array([[10,10,10]])
```

Expected Output:

```
Printing Original array
[[34 43 73]
 [82 22 12]
 [53 94 66]]

Array after deleting column 2 on axis 1
[[34 73]
 [82 12]
 [53 66]]
```

```
Array after inserting column 2 on axis 1
[[34 10 73]
 [82 10 12]
 [53 10 66]]
```

17) **Sort following NumPy array**
- **Case 1: Sort array by the second row**
- **Case 2: Sort the array by the second column**

```
sampleArray =
numpy.array([[34,43,73],[82,22,12],[53,94,66]])
```

Expected Output:

```
Printing Original array
[[34 43 73]
 [82 22 12]
 [53 94 66]]

Sorting Original array by second row
[[73 43 34]
 [12 22 82]
 [66 94 53]]

Sorting Original array by second column
[[82 22 12]
 [34 43 73]
 [53 94 66]]
```

18) **Create a result array by adding the following two NumPy arrays. Next, modify the result array by calculating the square of each element**

```
arrayOne = numpy.array([[5, 6, 9], [21 ,18, 27]])
arrayTwo = numpy.array([[15 ,33, 24], [4 ,7, 1]])
```

Expected Output:

```
addition of two arrays is

[[20 39 33]
 [25 25 28]]

Result array after calculating the square root of all
elements

[[ 400 1521 1089]
 [ 625  625  784]]
```

19) Write a NumPy program to compute the multiplication of two given matrixes.

20) Write a NumPy program to find a matrix or vector norm.

21) Write a NumPy program to compute the condition number of a given matrix.

22) Write a NumPy program to add, subtract, multiply, divide arguments element-wise.

23) Write a NumPy program to get the largest integer smaller or equal to the division of the inputs.

24) Write a NumPy program to get the powers of an array values element-wise.
Note: First array elements raised to powers from second array
Expected Output:
Original array
[0 1 2 3 4 5 6]
First array elements raised to powers from second array, element-wise:
[0 1 8 27 64 125 216]

25) Write a NumPy program to round array elements to the given number of decimals.

26) Write a NumPy program to multiply a 5x3 matrix by a 3x2 matrix and create a real matrix product.

27) Write a NumPy program to sort a given array of shape 2 along the first axis, last axis and on flattened array.

28) Write a NumPy program to create a structured array from given student name, height, class and their data types. Now sort by class, then height if class are equal.
Expected Output:
Original array:

[(b'James', 5, 48.5) (b'Nail', 6, 52.5) (b'Paul', 5, 42.1) (b'Pit', 5, 40.11)]
Sort by age, then height if class are equal:
[(b'Pit', 5, 40.11) (b'Paul', 5, 42.1) (b'James', 5, 48.5) (b'Nail', 6, 52.5)]

29) Write a NumPy program to sort an given array by the n^{th} column.
Original array:
[[1 5 0]
[3 2 5]
[8 7 6]]
Sort the said array by the nth column:
[[3 2 5]
[1 5 0]
[8 7 6]]

30) Write a NumPy program to sort the specified number of elements from beginning of a given array.
Sample output:
Original array:
[0.39536213 0.11779404 0.32612381 0.16327394 0.98837963 0.25510787 0.01398678 0.15188239 0.12057667 0.67278699]
Sorted first 5 elements:
[0.01398678 0.11779404 0.12057667 0.15188239 0.16327394 0.25510787 0.39536213 0.98837963 0.32612381 0.67278699]