

## Team Application Exercises (tAPP-4) - Solutions

### Task 1

Write a NumPy program to create an array of integers from 10 to 20. Write down its output when it is run.

```
import numpy as np
array=np.arange(10,20)
print("Array of the integers from 10 to 20")
print(array)
```

```
Array of the integers from 10 to 20
[10 11 12 13 14 15 16 17 18 19]
```

### Task 2:

Write a program that creates a 3x3 array. The program should the use `saveetxt()` and `loadtxt()` to save a given array to a text file and load it.

<pre>import numpy as np  x = np.arange(9).reshape(3,3) print("Original array:") print(x) np.savetxt('temp.txt', x, fmt="%d") print("After loading, content of the text file:") result = np.loadtxt('temp.txt') print(result)  # You can also do this y = np.arange(9) z = np.reshape(y, (3,3)) print(z) np.savetxt('temp1.txt', z, fmt="%d") print("After loading, content of the text file:") result = np.loadtxt('temp1.txt') print(result)</pre>	<pre>y = np.arange(9) print(y.reshape(3,3)) np.savetxt('temp1.txt', y, fmt="%d") print("After loading, content of the text file:") result = np.loadtxt('temp1.txt') print(result) print(result.reshape(3,3))</pre>
<p>Original array:</p> <pre>[[0 1 2]  [3 4 5]  [6 7 8]]</pre> <p>After loading, content of the text file:</p> <pre>[[0. 1. 2.]  [3. 4. 5.]  [6. 7. 8.]]</pre>	<pre>[[0 1 2]  [3 4 5]  [6 7 8]]</pre> <p>After loading, content of the text file:</p> <pre>[0. 1. 2. 3. 4. 5. 6. 7. 8.]</pre> <p>← output of print result</p> <pre>[[0. 1. 2.]  [3. 4. 5.]  [6. 7. 8.]]</pre> <p>← output of print result.reshape.</p> <p>Normally, in the correct solution, we only want to load what we saved and then print it. We do not want to reshape it again.</p> <p>Therefore the solution on the left is the best answer.</p>

<pre> x:=np.arange(200,209).reshape(3,-3) print("Original array:") print(x) np.savetxt('temp.txt',x,fmt="%d") print("After loading, content of the text file:") result=np.loadtxt('temp.txt') print(result)  # You can also do this y:=np.arange(12,24) z:=np.reshape(y,(4,3)) print(z) np.savetxt('temp1.txt',z,fmt="%d") print("After loading, content of the text file:") result=np.loadtxt('temp1.txt') print(result) </pre>	<pre> untitled0.py', wdir='C:/Users/mireilla/OneDrive - Original array: [[200 201 202]  [203 204 205]  [206 207 208]] After loading, content of the text file: [[200. 201. 202.]  [203. 204. 205.]  [206. 207. 208.]]  [[12 13 14]  [15 16 17]  [18 19 20]  [21 22 23]] After loading, content of the text file: [[12. 13. 14.]  [15. 16. 17.]  [18. 19. 20.]  [21. 22. 23.]] </pre>
--	--

### Task3:

Perform the following operations: Addition, subtraction and matrix product:

array1	array2
<div>[23, 45, 11]</div> <div>[12, 23, 54]</div> <div>[1, 23, 10]</div>	<div>[3, 5, 1]</div> <div>[2, 3, 4]</div> <div>[9, 1, 5]</div>

### Solutions

#### Addition of two arrays

Array1-row1	23	45	11
	+	+	+
Array2-row1	3	5	1
<b>Solution row1</b>	<b>26</b>	<b>50</b>	<b>12</b>

  

Array1-row2	12	23	54
	+	+	+
Array2-row2	2	3	4
<b>Solution row 2</b>	<b>14</b>	<b>26</b>	<b>58</b>

  

Array1-row3	1	23	10
	+	+	+
Array2-row3	9	1	5
<b>Solution row 3</b>	<b>10</b>	<b>24</b>	<b>15</b>

#### Subtraction

Array1-row1	23	45	11
	-	-	-
Array2-row1	3	5	1
<b>Solution row1</b>	<b>20</b>	<b>40</b>	<b>10</b>

  

Array1-row2	12	23	54
	-	-	-
Array2-row2	2	3	4
<b>Solution row 2</b>	<b>10</b>	<b>20</b>	<b>50</b>

  

Array1-row3	1	23	10
	-	-	-
Array2-row3	9	1	5
<b>Solution row 3</b>	<b>-8</b>	<b>22</b>	<b>5</b>

```

[[ 69 225 11]
 [ 24 69 216]
 [ 9 23 50]]

```

This would be the solution if were performing a simple matrix multiplication using the "\*" operator.

But this problem asked for the matrix product. So we will use numpy dot()

Addition	Subtraction
array([[26, 50, 12], [14, 26, 58], [10, 24, 15]])	array([[20, 40, 10], [10, 20, 50], [-8, 22, 5]])

### Matrix Product

<u>array1</u>	<u>array2</u>
[23, 45, 11] [12, 23, 54] [1, 23, 10]	[3, 5, 1] [2, 3, 4] [9, 1, 5]

- 1) To get the first row of our results, we use the **first row** of array1 and the **3 columns** of array2:

Array1 row **1** and array2 column 1:  $(23 * 3) + (45 * 2) + (11 * 9) = \mathbf{258}$

Array1 row **1** and array2 column 2:  $(23 * 5) + (45 * 3) + (11 * 1) = \mathbf{261}$

Array1 row **1** and array2 column 3:  $(23 * 1) + (45 * 4) + (11 * 5) = \mathbf{258}$

<b>258</b>	<b>261</b>	<b>258</b>

- 2) To get our second row of results, we use **row2** of array1 and **all 3 columns** of array2.

Array1 row **2** and array2 column 1:  $(12 * 3) + (23 * 2) + (54 * 9) = \mathbf{568}$

Array1 row **2** and array2 column 2:  $(12 * 5) + (23 * 3) + (54 * 1) = \mathbf{183}$

Array1 row **2** and array2 column 3:  $(12 * 1) + (23 * 4) + (54 * 5) = \mathbf{374}$

<b>258</b>	<b>261</b>	<b>258</b>
<b>568</b>	<b>183</b>	<b>374</b>

To get our last row of results, we use **row3** of array1 and **all 3 columns** of array2.

Array1 row **3** and array2 column 1:  $(1 * 3) + (23 * 2) + (10 * 9) = 139$

Array1 row **3** and array2 column 2:  $(1 * 5) + (23 * 3) + (10 * 1) = 84$

Array1 row **3** and array2 column 3:  $(1 * 1) + (23 * 4) + (10 * 5) = 143$

<b>258</b>	<b>261</b>	<b>258</b>
<b>568</b>	<b>183</b>	<b>374</b>
<b>139</b>	<b>84</b>	<b>143</b>

```
[[258 261 258]
 [568 183 374]
 [139  84 143]]
```

So the results of our matrix dot product is:

#### **Task4:**

Perform the following operations: Addition, subtraction and matrix product:

<u>array1</u>	<u>array2</u>
[3, 5, 1] [2, 3, 5] [1, 3, 1]	[8, 5, 1] [2, 9, 4] [2, 10, 5]

#### Solutions

<u>Addition</u>	<u>Subtraction</u>	<u>Matrix product</u>
array([[11, 10, 2], [ 4, 12, 9], [ 3, 13, 6]])	array([[ -5,  0,  0], [  0, -6,  1], [ -1, -7, -4]])	array([[36, 70, 28], [32, 87, 39], [16, 42, 18]])

<u>array1</u>	<u>array2</u>
[4, 7, 2] [2, 3, 2] [2, 3, 1]	[3, 5, 1] [5, 7, 4] [2, 5, 5]