

PYTHON MODULE - 5

1.Create a numPy program for the following problem statement.

You are given a 2-D array with dimensions N X M.

Your task is to perform the *sum* tool over axis 0 and then find the product of the result.

For example:

For example:

Input	Result
2 2 1 2 3 4	24

PROGRAM:

```
import numpy as np
n,m=map(int, input().split())
arr=[]
for _ in range(n):
    r=list(map(int, input().split()))
    arr.append(r)
n_array=np.array(arr)
summ=np.sum(n_array,axis=0)
pr=np.prod(summ)
print(pr)
```

RESULT:

	Input	Expected	Got	
✓	2 2 1 2 3 4	24	24	✓
✓	2 2 1 2 3 2	16	16	✓
✓	3 3 1 1 1 1 1 1 1 1 1	27	27	✓

Passed all tests! ✓

2.Create a numpy program for the following problem statement.

Array re-shaping

Convert a 1-D array to a 2-D Array 2 2X2 arrays

input the number of elements,rows,columns and number of arrays

For example:

Input	Result
8 2 2 2	[[[0 1] [2 3]] [[4 5] [6 7]]]

PROGRAM:

```
import numpy as np
a=int(input())
b=int(input())
c=int(input())
d=int(input())
arr=np.arange(a)
re_arr=arr.reshape(b,c,d)
print(re_arr)
```

RESULT:

	Input	Expected	Got	
✓	8 2 2 2	[[[0 1] [2 3]] [[4 5] [6 7]]]	[[[0 1] [2 3]] [[4 5] [6 7]]]	✓
✓	27 3 3 3	[[[0 1 2] [3 4 5] [6 7 8]] [[9 10 11] [12 13 14] [15 16 17]] [[18 19 20] [21 22 23] [24 25 26]]]	[[[0 1 2] [3 4 5] [6 7 8]] [[9 10 11] [12 13 14] [15 16 17]] [[18 19 20] [21 22 23] [24 25 26]]]	✓

Passed all tests! ✓

3.Create a numPy program for the following problem statement.
You are given two integer arrays A and B of Dimensions N and M.

Your task is to perform the following operations:

- 1.add(A+B)
- 2.Subtract(A-B)
- 3.Multiply(A*B)
- 4.Integer Division(A/B)
- 5.Modulo(A%B)
- 6.Power(A**B)

For example:

Input	Result
1 4	[[6 8 10 12]]
1 2 3 4	[[-4 -4 -4 -4]]
5 6 7 8	[[5 12 21 32]]
	[[0 0 0 0]]
	[[1 2 3 4]]
	[[1 64 2187
	65536]]

PROGRAM:

```
import numpy as np
```

```
dims=input().split()
```

```
n,m=int(dims[0]),int(dims[1])
```

```
c=np.array([list(map(int,input().split()))for _ in range(n)])
```

```
d=np.array([list(map(int,input().split()))for _ in range(n)])
```

```
print(np.add(c,d))
```

```
print(np.subtract(c,d))
```

```

print(np.multiply(c,d))
print(np.floor_divide(c,d))
print(np.mod(c,d))
print(np.power(c,d))

```

RESULT:

	Input	Expected	Got	
✓	1 4 1 2 3 4 5 6 7 8	[[6 8 10 12]] [[-4 -4 -4 -4]] [[5 12 21 32]] [[0 0 0 0]] [[1 2 3 4]] [[1 64 2187 65536]]	[[6 8 10 12]] [[-4 -4 -4 -4]] [[5 12 21 32]] [[0 0 0 0]] [[1 2 3 4]] [[1 64 2187 65536]]	✓
✓	1 4 1 2 3 4 5 6 7 7	[[6 8 10 11]] [[-4 -4 -4 -3]] [[5 12 21 28]] [[0 0 0 0]] [[1 2 3 4]] [[1 64 2187 16384]]	[[6 8 10 11]] [[-4 -4 -4 -3]] [[5 12 21 28]] [[0 0 0 0]] [[1 2 3 4]] [[1 64 2187 16384]]	✓
✓	2 4 1 2 3 4 1 2 3 4 5 6 7 7 5 6 7 7	[[6 8 10 11]] [6 8 10 11]] [[-4 -4 -4 -3]] [-4 -4 -4 -3]] [[5 12 21 28]] [5 12 21 28]] [[0 0 0 0]] [0 0 0 0]] [[1 2 3 4]] [1 2 3 4]] [[1 64 2187 16384]] [1 64 2187 16384]]	[[6 8 10 11]] [6 8 10 11]] [[-4 -4 -4 -3]] [-4 -4 -4 -3]] [[5 12 21 28]] [5 12 21 28]] [[0 0 0 0]] [0 0 0 0]] [[1 2 3 4]] [1 2 3 4]] [[1 64 2187 16384]] [1 64 2187 16384]]	✓

Passed all tests! ✓

4.Create a pandas program to get the positions of items of series A in another series B?

For example:

Input	Result
[10,9,6,5,3,1,12,8,13] [1,3,10,13]	[5, 4, 0, 8]

PROGRAM:

```
import pandas as pd
a=eval(input())
b=eval(input())
s_a=pd.Series(a)
s_b=pd.Series(b)
pos=[s_a[s_a==item].index[0] for item in s_b]
print(pos)
```

RESULT:

	Input	Expected	Got	
✓	[10,9,6,5,3,1,12,8,13] [1,3,10,13]	[5, 4, 0, 8]	[5, 4, 0, 8]	✓
✓	[20,6,7,4,2,1,13,91,13] [6,4,2,91]	[1, 3, 4, 7]	[1, 3, 4, 7]	✓

Passed all tests! ✓

5. Create a Pandas program to append rows to an existing DataFrame and display the combined data.

For example:

Input	Result
<pre>{'id':['S1','S2','S3','S4','S5'],'name':['Danni','Ryder','Bryce','Ber','Morin'],'marks':[200, 210, 190, 222, 199]} ['S6','John',90]</pre>	<pre>Original Dataframe id name marks 0 S1 Danni 200 1 S2 Ryder 210 2 S3 Bryce 190 3 S4 Ber 222 4 S5 Morin 199 combined Dataframe id name marks 0 S1 Danni 200 1 S2 Ryder 210 2 S3 Bryce 190 3 S4 Ber 222 4 S5 Morin 199 5 S6 John 90</pre>

PROGRAM:

```
import pandas as pd
a=eval(input())
b=eval(input())
s_a=pd.DataFrame(a)
print("Original Dataframe")
print("",s_a)
r=pd.DataFrame([b],columns=s_a.columns)
res=pd.concat([s_a,r],ignore_index="True")
print("combined Dataframe")
print("",res)
```

RESULT:

	Expected	Got	
'S5'], 'name': ['Danni', 'Ryder', 'Bryce', 'Ber', 'Morin'], 'marks': [200, 210, 190, 222, 199]}	Original Dataframe <pre> id name marks 0 S1 Danni 200 1 S2 Ryder 210 2 S3 Bryce 190 3 S4 Ber 222 4 S5 Morin 199 </pre> combined Dataframe <pre> id name marks 0 S1 Danni 200 1 S2 Ryder 210 2 S3 Bryce 190 3 S4 Ber 222 4 S5 Morin 199 5 S6 John 90 </pre>	Original Dataframe <pre> id name marks 0 S1 Danni 200 1 S2 Ryder 210 2 S3 Bryce 190 3 S4 Ber 222 4 S5 Morin 199 </pre> combined Dataframe <pre> id name marks 0 S1 Danni 200 1 S2 Ryder 210 2 S3 Bryce 190 3 S4 Ber 222 4 S5 Morin 199 5 S6 John 90 </pre>	✓
'name': ['sam', 'john', 'siva'], 'designation': ['Manager', 'clerk', 'cashier']}]}	Original Dataframe <pre> e_id name designation 0 e1 sam Manager 1 e2 john clerk 2 e3 siva cashier </pre> combined Dataframe <pre> e_id name designation 0 e1 sam Manager 1 e2 john clerk 2 e3 siva cashier 3 e4 ram accountant </pre>	Original Dataframe <pre> e_id name designation 0 e1 sam Manager 1 e2 john clerk 2 e3 siva cashier </pre> combined Dataframe <pre> e_id name designation 0 e1 sam Manager 1 e2 john clerk 2 e3 siva cashier 3 e4 ram accountant </pre>	✓