1) Arrays

In [1]:

```
import numpy

def arrays(arr):
    return(numpy.array(arr[::-1], float))

arr = input().strip().split(' ')
result = arrays(arr)
print(result)
```

```
1 2 3 4 -8 -10
[-10.  -8.   4.   3.   2.   1.]
```

2) Shape and Reshape

In [2]:

```
import numpy
arr = list(map(int,input().split()))
numpy_array = numpy.array(arr)
print(numpy.reshape(numpy_array,(3,3)))
```

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

3) Transpose and Flatten

In [5]:

```
n, m = map(int, input().split())
ans = numpy.array([input().strip().split() for _ in range(n)], int)
print(ans.transpose())
print(ans.flatten())
```

```
1 2
2 3
[[2]
 [3]]
[2 3]
```

4) Concatenate

In [7]:

```python
import numpy as np
a, b, c = map(int,input().split())
arrA = np.array([input().split() for i in range(a)],int)
arrB = np.array([input().split() for i in range(b)],int)
print(np.concatenate((arrA, arrB), axis = 0))
```

```
4 3 2
1 2
1 2
1 2
1 2
3 4
3 4
3 4
[[1 2]
 [1 2]
 [1 2]
 [1 2]
 [3 4]
 [3 4]
 [3 4]]
```

5) Zeros and Ones

In [8]:

```python
import numpy as np
ans = tuple(map(int,input().split()))
print(np.zeros(ans,int), np.ones(ans,int), sep = '\n')
```

```
3 3 3
[[[0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]]

 [[0 0 0]
  [0 0 0]
  [0 0 0]]]
[[[1 1 1]
  [1 1 1]
  [1 1 1]]

 [[1 1 1]
  [1 1 1]
  [1 1 1]]

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

6) Eye and Identity

In [9]:

```
import numpy
numpy.set_printoptions(sign=' ')
print(numpy.eye(*map(int, input().split())))
```

```
3 3
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]]
```

7) Array Mathematics

In [10]:

```
import numpy
n,m = map(int,input().split())
a = numpy.array([input().split() for i in range(n)],int)
b = numpy.array([input().split() for i in range(n)],int)
print(a+b,a-b,a*b,a//b,a%b,a**b,sep="\n")
```

```
2 5
1 2 3 4
5 6 7 8
2 3 4 5
2 3 4 5
[[ 3  5  7  9]
 [ 7  9 11 13]]
[[-1 -1 -1 -1]
 [ 3  3  3  3]]
[[ 2  6 12 20]
 [10 18 28 40]]
[[0 0 0 0]
 [2 2 1 1]]
[[1 2 3 4]
 [1 0 3 3]]
[[    1     8    81  1024]
 [   25   216  2401 32768]]
```

8) Floor, Ceil and Rint

In [15]:

```
import numpy

numpy.set_printoptions(sign=' ')

array = numpy.array(input().split(),float)

print(numpy.floor(array))
print(numpy.ceil(array))
print(numpy.rint(array))
```

```
1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 9.9
[ 1.  2.  3.  4.  5.  6.  7.  8.  9.]
[  2.   3.   4.   5.   6.   7.   8.   9.  10.]
[  1.   2.   3.   4.   6.   7.   8.   9.  10.]
```

9) Sum and Prod

In [16]:

```
import numpy
A = []

for i in range(int(input().split(' ')[0])):
    A.append([int(x) for x in input().split(' ')])
A = numpy.array(A)

sum_prod = numpy.sum(A, axis=0)
sum_prod = numpy.prod(sum_prod)

print(sum_prod)
```

```
2 2
1 2
3 4
24
```

10) Min and Max

In [17]:

```
import numpy

n,m=map(int,input().split())

ans =[list(map(int,input().split())) for i in range(n)]

arr=numpy.array(ans)

print(max(numpy.min(arr,axis=1)))
```

```
4 2
2 5
3 7
1 3
4 0
3
```

11) Mean, Var, and Std

In [18]:

```
import numpy as np


n, m = [int(x) for x in input().strip().split()]


arr = np.array([[int(x) for x in input().strip().split()] for _ in range(n)]


print(np.mean(arr, axis = 1))
print(np.var(arr, axis = 0))
print(np.round(np.std(arr), decimals = 11))
```

```
2 2
1 2
3 4
[ 1.5  3.5]
[ 1.  1.]
1.11803398875
```

12) Dot and Cross

In [19]:

```
import numpy
n = int(input())
a = numpy.array([input().split() for _ in range(n)], int)
b = numpy.array([input().split() for _ in range(n)], int)
print(numpy.dot(a, b))
```

```
2
1 2
3 4
1 2
3 4
[[ 7 10]
 [15 22]]
```

13) Inner and Outer

In [20]:

```
import numpy

A = numpy.array(input().split(), int)
B = numpy.array(input().split(), int)
print(numpy.inner(A, B), numpy.outer(A, B), sep='\n')
```

```
0 1
2 3
3
[[0 0]
 [2 3]]
```

14) Polynomials

In [22]:

```
import numpy as np
poly = [float(x) for x in input().split()]
x = float(input())
print(np.polyval(poly, x))
```

```
1.1 2 3
0
3.0
```

15) Linear Algebra

In [23]:

```
import numpy
array=[list(map(float,input().split())) for i in range(int(input()))]
print(round(numpy.linalg.det(array),2))
```

```
2
1.1 1.1
1.1 1.1
0.0
```