In [1]:

```
#Question 1: Arrays

import numpy

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

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

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

In [2]:

```
#Question 2: Shape and Reshape

import numpy as np
array=np.array(list(map(int,input().split())))
array.shape=(3,3)
print(array)
```

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

In [3]:

```
#Question 3: Transpose and Flatten

import numpy


N,M=map(int,input().split())

lst=[list(map(int,input().split())) for _ in range(N)]
x=numpy.array(lst)

print(numpy.transpose(x))
print(x.flatten())
```

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

In [5]:

```
#Question 4: Concatenate

import numpy


N,M,P=map(int,input().split())

lst1=[list(map(int,input().split())) for i in range(N)]
lst2=[list(map(int,input().split())) for i in range(M)]
x1=numpy.array(lst1)
x2=numpy.array(lst2)

print(numpy.concatenate((x1,x2),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]]
```

In [6]:

```
#Question 5: Zeros and Ones

import numpy


N = tuple(map(int, input().split()))
print(numpy.zeros(N, dtype = numpy.int), numpy.ones(N, dtype = numpy.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]]]
<ipython-input-6-375601aae91d>:7: DeprecationWarning: `np.int` is a deprecated alias for the builtin `int`. To silence this warning, use `int` by itself. Doing this will not modify any behavior and is safe. When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If you wish to review your current use, check the release note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  print(numpy.zeros(N, dtype = numpy.int), numpy.ones(N, dtype = numpy.int), sep='\n')
```

In [7]:

```
#Question 6: Eye and Identity

import numpy


print(str(numpy.eye(*map(int,input().split()))).replace('1',' 1').replace('0',' 0'))
```

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

In [8]:

```
#Question 7: Array Mathemetics

import numpy


N, M = tuple(map(int, input().split()))

a = numpy.array([input().split() for i in range(0, N)], int)
b = numpy.array([input().split() for j in range(0, N)], int)

print(a+b, a-b, a*b, a//b, a % b, a**b, sep='\n')
```

```
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]]
```

In [9]:

```
#Question 8: Floor,Ceil and Rint

import numpy



numpy.set_printoptions(sign=' ')

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

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

```
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.]
```

In [10]:

```
#Question 9: Sum and Prod

N, M = tuple(map(int, input().split()))
lst = []
for i in range(0, N):
    lst.append(input().split())

arr = numpy.array(lst, int)
print(numpy.prod(numpy.sum(arr, axis=0), axis=None))
```

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

In [11]:

```
#Question 10: Min and Max

N, M = map(int, input().split())
lst = []
for i in range(0, N):
    lst.append([x for x in input().split()])
```

```
arr = numpy.array(lst, int)
print(numpy.max(numpy.min(arr, axis=1), axis=None))
```

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

In [1]:

```
#Question 11: Mean,Var,Std

import numpy
numpy.set_printoptions(sign=' ')
N, M = map(int, input().split())
arr = numpy.array([input().split() for _ in range(N)], float)
print(numpy.mean(arr, axis = 1))
print (numpy.var(arr, axis = 0))
print(numpy.around(numpy.std(arr), 12))
```

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

In [2]:

```
#Question 12: Dot and Cross

import numpy
N = int(input())
lst1, lst2 = [], []
for i in range(0, N):
    lst1.append([x for x in input().split()])

for j in range(0, N):
    lst2.append([y for y in input().split()])

arr1 = numpy.array(lst1, int)
arr2 = numpy.array(lst2, int)

#  we  use matmul() instead of the traditional dot/cross
print(numpy.matmul(arr1, arr2))
```

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

In [3]:

```
#Question 13: Inner and Outer

import numpy


a1 = numpy.array(input().split(), int)
a2 = numpy.array(input().split(), int)

print(numpy.inner(a1,a2), numpy.outer(a1,a2), sep='\n')
```

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

```
 [2 3]]
```

In [4]:

```
#Question 14: Polynomials

import numpy


P = [float(x) for x in input().split()]
value = float(input())

print(numpy.polyval(P, value))
```

```
1.1 2 3
0
3.0
```

In [5]:

```
#Question 15: Linear Algebra

import numpy

n=int(input())
arr=numpy.array([input().split() for _ in range(n)],float)
numpy.set_printoptions(legacy='1.13')
print(numpy.linalg.det(arr))
```

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

In [ ]:

```

```