

[10장 과제]

맑은 고딕 12 ☞ 학번 : 20203103 이름 : 임정민

실습 10-1 (p. 252 도전문제 10-1) 다음과 같이 a, b의 값을 출력하고, 각각의 연산 결과를 화면과 같이 출력하시오.

[결과 화면]

```
python main.py
a= [ 1  2  3  4  5  6  7  8  9 10] b= [ 10  20  30  40  50  60  70  80  90 100]
a+b = [ 11  22  33  44  55  66  77  88  99 110]
a-b = [ -9 -18 -27 -36 -45 -54 -63 -72 -81 -90]
a*b = [ 10  40  90 160 250 360 490 640 810 1000]
a/b = [0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]
```

[프로그램 소스]

```
##challenge 10-1 (252p)
import numpy as np
a = np.array(range(1,11))
b = np.array(range(10, 101, 10))
```

```
print("a=", a, "b=", b)
print("a+b =",a+b)
print("a-b =",a-b)
print("a*b =",a*b)
print("a/b =",a/b)
```

실습 10-2 (p.254, LAB10-1) LAB10-1에 주어진 내용을 이해하고, 프로그램을 입력하여 다음 [결과 화면]과 같이 결과를 출력하시오.

[결과 화면]

```
python main.py
실습 1 : [0 1 2 3 4 5 6 7 8 9]
실습 2 : [0 1 2 3 4 5 6 7 8 9]
실습 3 : [0 2 4 6 8]
실습 4 :
array_c의 shape : (5,)
array_c의 ndim : 1
array_c의 dtype : int64
array_c의 size : 5
array_c의 itemsize : 8
```

[프로그램 소스]

```
##lab 10-1.py
import numpy as np
array_a = np.array([0,1,2,3,4,5,6,7,8,9])
array_b = np.array(range(0,10))
array_c = np.array(range(0,10,2))
print("실습 1 :",array_a)
print("실습 2 :",array_b)
print("실습 3 :",array_c)
```

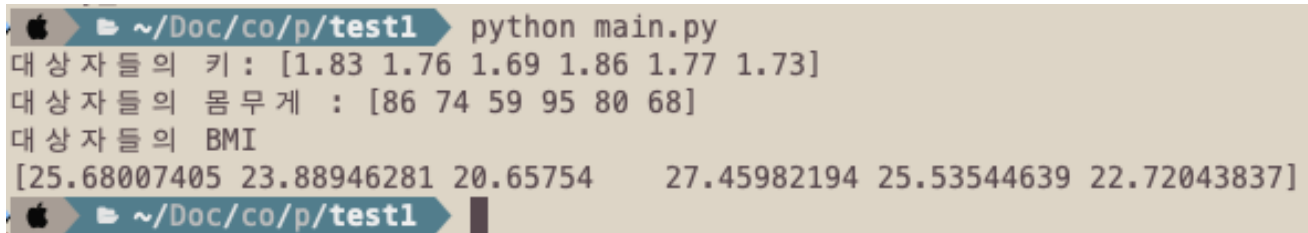
```

print("실습 4 :")
print('array_c 의 shape :', array_c.shape)
print('array_c 의 ndim :', array_c.ndim)
print('array_c 의 dtype :', array_c.dtype)
print('array_c 의 size :', array_c.size)
print('array_c 의 itemsize :', array_c.itemsize)

```

실습 10-3 (p.256 LAB10-2 응용) LAB10-2를 이해하고, 다음 [결과 화면]과 같이 결과가 출력되도록 프로그램을 수정하십시오.

[결과]



```

python main.py
대상자들의 키 : [1.83 1.76 1.69 1.86 1.77 1.73]
대상자들의 몸무게 : [86 74 59 95 80 68]
대상자들의 BMI
[25.68007405 23.88946281 20.65754      27.45982194 25.53544639 22.72043837]

```

[프로그램 소스]

```

##lab10-2.py
import numpy as np

heights = [1.83, 1.76, 1.69, 1.86, 1.77, 1.73]
weights = [86, 74, 59, 95, 80, 68]

np_heights = np.array(heights)
np_weights = np.array(weights)

bmi = np_weights/(np_heights**2)

print('대상자들의 키:', np_heights)
print('대상자들의 몸무게 :', np_weights)
print('대상자들의 BMI')
print(bmi)

```

실습 10-4 (p.264 LAB10-4 응용) LAB10-4를 이해하고, 다음 [결과 화면]과 같이 결과가 출력되도록 프로그램을 수정하시오. (책의 프로그램과 결과에 오류 있음, 수정 요망)

[결과 화면]

```
python main.py
x : [[ 1.83  1.76  1.69  1.86  1.77  1.73]
      [86.   74.   59.   95.   80.   68.  ]]
y : [[ 1.76  1.69]
      [74.   59.  ]]
z : [[86. 74. 59. 95. 80. 68.]]
x shape : (2, 6)
y shape : (2, 2)
z shape : (1, 6)
z values = : [86. 74. 59. 95. 80. 68.]
BMI data
[25.68007405 23.88946281 20.65754      27.45982194 25.53544639 22.72043837]
```

[프로그램 소스]

```
##lab10-4.py

import numpy as np

x = np.array([[1.83,1.76,1.69,1.86,1.77,1.73],
              [86.0,74.0,59.0,95.0,80.0,68.0]])

y = x[0:2,1:3]
z = x[0:2][1:3]
print('x : ', x)
print('y : ', y)
print('z : ', z)
print('x shape :', x.shape)
print('y shape :', y.shape)
print('z shape :', z.shape)
print('z values = :', z)

bmi = x[1] / x[0]**2

print ('BMI data')
print(bmi)
```

실습 10-5 (p.265 LAB10-5 응용) LAB10-5를 이해하고, 선수 정보에 나이(23, 24, 25, 26)를 차례로 추가 하여, 다음 [결과 화면]과 같이 결과가 출력되도록 프로그램을 수정하시오.

[결과 화면]

```
python main.py
몸무게가 80 이상인 선수 정보
[[183.  86.2  24. ]
 [176.  80.1  26. ]]
키가 180 이상인 선수 정보
[[183.  86.2  24. ]
 [181.  78.5  25. ]]
나이가 25 이상인 선수 정보
[[170.  76.4  23. ]
 [183.  86.2  24. ]
 [181.  78.5  25. ]]
```

[프로그램 소스]

##lab10-5.py

```
import numpy as np
```

```
players = [[170,76.4,23],
            [183,86.2,24],
            [181,78.5,25],
            [176,80.1,26]]
```

```
np_players = np.array(players)
```

```
print('몸무게가 80 이상인 선수 정보')
print(np_players[np_players[:,1] >= 80.0])
```

```
print('키가 180 이상인 선수 정보')
print(np_players[np_players[:,0] >= 180.0])
```

```
print('나이가 25 이상인 선수 정보')
print(np_players[np_players[:,2] <=25])
```

실습 10-6 (p.273 LAB10-6 응용) LAB10-6을 이해하고, 출력 정보에 신장 총계, 체중 총계, 나이 총계를 추가하여, 다음 [결과 화면]과 같이 결과가 출력되도록 프로그램을 수정하시오. 단, 모든 총계는 소숫점 없이 정수값만 출력되도록 한다.

[결과 화면]

🍏 ~/Doc/co/p/test1 python main.py

```
신장 평균값 : 175.8792818477416
신장 중앙값 : 176.44080192415646
신장 총계 17588
체중 평균값 : 69.20519657659875
체중 중간값 : 68.74313254289187
체중 총계 6921
나이 평균값 : 22.19
나이 중간값 : 23.5
나이 총계 : 2219
```

🍏 ~/Doc/co/p/test1

[프로그램 소스]

```
##lab10-6.py

import numpy as np

players = np.zeros((100,3))

players[:,0] = 10 * np.random.randn(100) + 175
players[:,1] = 10 * np.random.randn(100) + 70
players[:,2] = np.floor(10*np.random.randn(100)) + 22

heights = players[:,0]
print('신장 평균값:', np.mean(heights))
print('신장 중앙값:', np.median(heights))
print('신장 총계', round(np.sum(heights)))

weights = players[:,1]
print('체중 평균값:', np.mean(weights))
print('체중 중간값:', np.median(weights))
print('체중 총계', round(np.sum(weights)))

ages = players[:,2]
print('나이 평균값:', np.mean(ages))
print('나이 중간값:', np.median(ages))
print('나이 총계:', round(np.sum(ages)))
```

실습 10-7 (p.277 심화문제 10.2 응용) 다음 0에서 24까지의 값을 가지는 5×5 크기의 2차원 행렬 n_arr
를 생성한 후 [결과 화면]과 같이 결과가 출력되도록 프로그램을 작성하시오.

[결과 화면]

python main.py

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

2)

첫 원소 : 0

마지막 원소 : 24

3)

```
[[0 1 2 3 4]
 [5 6 7 8 9]]
```

4)

```
[[10 11 12 13 14]
 [15 16 17 18 19]
 [20 21 22 23 24]]
```

5)

```
[[ 0  2  4]
 [ 5  7  9]
 [10 12 14]
 [15 17 19]
 [20 22 24]]
```

6)

```
[[ 0  2  4]
 [10 12 14]
 [20 22 24]]
```

7)

```
[[0 1]
 [2 3]
 [4 5]
 [6 7]
 [8 9]]
```

[프로그램 소스]

```
##ex10-2.py
```

```
import numpy as np
```

```
n_arr = np.array([[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]])
```

```
#1)
```

```
print('1)', n_arr)
```

```
#2)
```

```
print('2)')
```

```
print('첫 원소 :', n_arr[0,0])
```

```
print('마지막 원소 :', n_arr[-1,-1])
```

```
#3)
```

```
print('3)')
```

```

n_arr2 = n_arr[:2]
print(n_arr2)

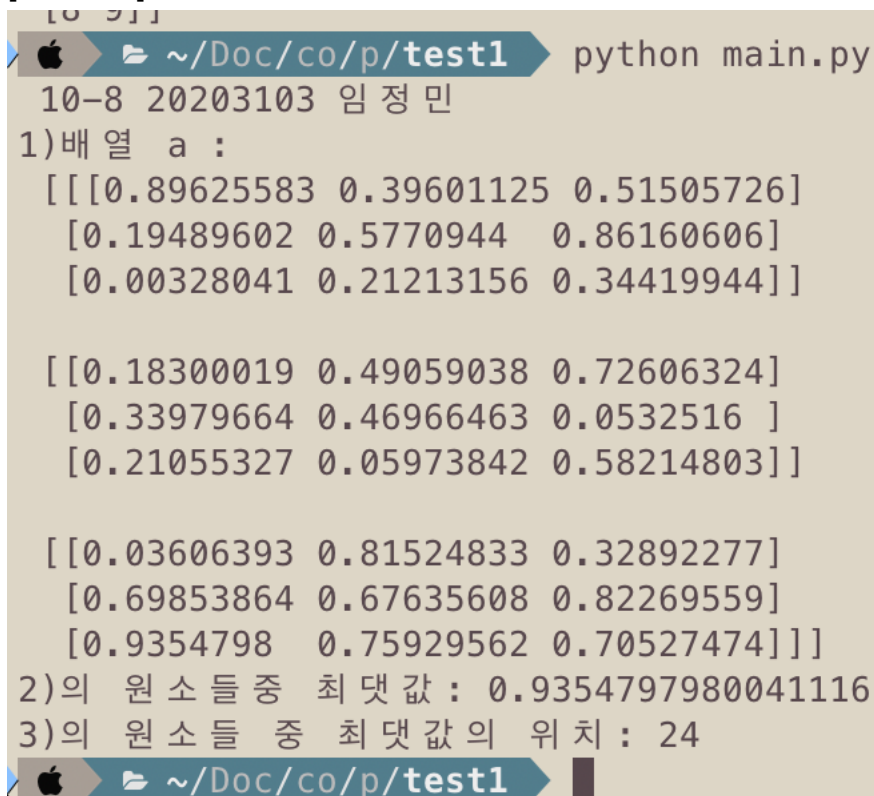
#4)
print('4')
n_arr3 = n_arr[2:]
print(n_arr3)

#5)
print('5')
n_arr4 = n_arr[0:,:2]
print(n_arr4)
#6)
print('6')
n_arr5 = n_arr[:,::2]
print(n_arr5)
#7)
print('7')
n_arr6 = n_arr[:2]
n_arr6 = np.reshape(n_arr6, (5,2))
print(n_arr6)

```

실습 10-8 (p.278 심화문제 10.3 응용) [결과 화면]과 같이 3×3×3 형태의 배열 a를 생성한 후 [결과 화면]과 같이 결과가 출력되도록 프로그램을 작성하시오.

[결과 화면]



```

10-8 20203103 임정민
1)배열 a :
[[[0.89625583 0.39601125 0.51505726]
  [0.19489602 0.5770944 0.86160606]
  [0.00328041 0.21213156 0.34419944]]

 [[0.18300019 0.49059038 0.72606324]
  [0.33979664 0.46966463 0.0532516 ]
  [0.21055327 0.05973842 0.58214803]]

 [[0.03606393 0.81524833 0.32892277]
  [0.69853864 0.67635608 0.82269559]
  [0.9354798 0.75929562 0.70527474]]]
2)의 원소들 중 최댓값 : 0.9354797980041116
3)의 원소들 중 최댓값의 위치 : 24

```

[프로그램 소스]

```
#ex10-3.py
import numpy as np

print(' 10-8 20203103 임정민')
array_a = np.random.rand(3,3,3)
print('1)배열 a : \n',array_a)

print('2)의 원소들중 최댓값:', np.max(array_a))
print('3)의 원소들 중 최댓값의 위치:',np.argmax(array_a))
```


실습 10-9 (p.278 심화문제 10.5 응용) [결과 화면]과 같이 4×4×2 형태의 배열 a를 생성한 후 [결과 화면]과 같이 결과가 출력되도록 프로그램을 작성하시오.

[결과 화면]

```
python main.py
10-9 20203103 임정민
[[ 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 27]
 [28 29]
 [30 31]]]
a : [[ 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 27]
 [28 29]
 [30 31]]]
b : [ 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 27 28 29 30 31]
10번째 원소 : 9
20번째 원소 : 19
```

[프로그램 소스]

```
#ex10-5.py
import numpy as np

print('10-9 20203103 임정민')
a = np.arange(0,32).reshape(4,4,2)

print(a)

b = a.flatten()

print('a : ',a)
print('b : ', b)
print('10 번째 원소 : ', b[9])
```

```
print('20 번째 원소 :', b[19])
```

실습 10-10 (p.279 심화문제 10.6 응용) 문제에 주어진 4개의 리스트 데이터를 [결과 화면]과 같이 생성하여 출력한 후 상관 관계를 출력하는 프로그램을 작성하시오.

[결과 화면]

```
python main.py
10-10 20203103 임정민
x1 :
[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, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
x2:
[6, 8, 3, 11, 9, 13, 11, 15, 11, 17, 14, 16, 19, 22, 20, 24, 22, 23, 22, 20, 27, 29
, 27, 27, 33, 33, 35, 32, 32, 30, 37, 33, 34, 42, 35, 39, 38, 40, 42, 43, 43, 50, 50
, 49, 53, 53, 55, 52, 57, 50, 56, 56, 56, 56, 57, 60, 61, 60, 67, 64, 61, 70, 65, 65
, 73, 67, 67, 75, 71, 74, 72, 80, 73, 77, 83, 76, 83, 84, 84, 80, 81, 89, 85, 85, 92
, 86, 94, 94, 93, 98, 97, 100, 98, 95, 96, 100, 102, 103, 103]
x3:
[44, 17, 30, 19, 40, 33, 8, 14, 40, 23, 32, 33, 59, 58, 52, 48, 26, 55, 29, 50, 32,
33, 41, 56, 54, 72, 55, 35, 34, 57, 75, 36, 35, 36, 71, 38, 68, 81, 71, 70, 64, 43,
72, 84, 70, 87, 74, 63, 54, 96, 51, 61, 58, 94, 98, 101, 74, 85, 60, 92, 74, 98, 63
, 68, 103, 88, 111, 77, 85, 113, 84, 104, 112, 93, 111, 123, 95, 114, 109, 93, 116,
104, 95, 91, 121, 94, 103, 130, 118, 117, 105, 130, 103, 142, 122, 102, 142, 117, 14
1, 121]
x4:
[46, 79, 72, 10, 20, 50, 64, 90, 84, 65, 86, 80, 102, 24, 76, 76, 29, 28, 66, 52, 5
8, 99, 78, 112, 39, 115, 107, 61, 49, 48, 84, 124, 109, 124, 97, 132, 49, 57, 88, 57
, 92, 133, 92, 88, 94, 116, 74, 72, 86, 132, 62, 118, 83, 68, 136, 56, 125, 92, 142,
140, 118, 129, 67, 78, 147, 133, 146, 143, 157, 80, 121, 77, 114, 121, 82, 147, 119
, 149, 167, 91, 111, 148, 174, 128, 177, 90, 98, 124, 94, 160, 96, 187, 166, 158, 15
3, 146, 164, 100, 180, 148]
[[1.          0.99567361 0.90210192 0.6964772 ]
 [0.99567361 1.          0.89850061 0.69848484]
 [0.90210192 0.89850061 1.          0.63812571]
 [0.6964772  0.69848484 0.63812571 1.          ]]
```

[프로그램 소스]

```
#10-6.py
import numpy as np

print('10-10 20203103 임정민')

x1 = [i for i in range(100)]
x2 = [i+ np.random.randint(1,10) for i in range(100)]
x3 = [i+ np.random.randint(1,50) for i in range(100)]
x4 = [i+ np.random.randint(1,100) for i in range(100)]
```

```
result = np.corrcoef([x1,x2,x3,x4])

print('x1 :\n',x1)
print('x2:\n',x2)
print('x3:\n',x3)
print('x4:\n',x4)

print(result)
```