

인공지능 Homework #1

01 다음 코드의 최종 결과를 쓰시오.

```
>>> import numpy as np
>>> my_vector = np.array([1, 2, 3, 4, 5, 6])
>>> selection = my_vector % 2 == 0
>>> my_vector[selection]
```

HINT 넘파이 배열에 연산자를 적용하면 배열의 요소마다 연산자가 적용된다.

02 다음 코드의 최종 결과를 쓰시오.

```
>>> import numpy as np
>>> first_matrix = np.array([[1, 2, 3], [4, 5, 6]])
>>> second_matrix = np.array([1, 2, 3])
>>> first_matrix + second_matrix
```

HINT 넘파이 배열끼리 연산을 할 때, 크기가 다르면 넘파이는 자동으로 배열의 크기를 확장한다. 이것을 브로드캐스팅 기능이라고 한다.

03 크기가 10인 널 벡터를 생성하고 다섯 번째 요소는 1로 설정하는 코드를 작성하라.

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

HINT np.zeros(10)를 사용하면 크기가 10인 널 벡터를 생성할 수 있다.

04 10에서 19까지의 값을 가지는 1차원 배열을 생성하라.

```
[10 11 12 13 14 15 16 17 18 19]
```

HINT np.arange()를 사용해본다.

05 0부터 9까지의 값으로 넘파이 1차원 배열을 채우고, 이 배열을 거꾸로 하는 문장을 작성하라.

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

HINT 슬라이싱 array[::-1]을 사용하면 배열의 요소들의 순서를 거꾸로 할 수 있다.

06 0부터 8까지의 값을 가지고 크기가 3×3인 행렬을 생성하라.

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

HINT reshape(3, 3)을 넘파이 배열에 적용한다.

07 난수로 채워진 3x3 넘파이 배열을 생성해보자.

```
[[0.94479439 0.18830511 0.76377723]
 [0.07878508 0.42415636 0.88907331]
 [0.28866152 0.02901134 0.64383825]]
```

HINT np.random.random()을 사용해본다.

08 임의의 값으로 10x10 배열을 만들고 최소값과 최대값을 찾아보자.

```
최소값 = 0.004397914499617905 최대값 = 0.9941731976328123
```

HINT np.random.random()과 min(), max()를 사용해본다.

09 배열의 테두리에 1, 내부에 0을 가진 3×3 크기의 2차원 배열을 생성해보자. 슬라이싱을 이용한다.

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

HINT A[1:-1,1:-1] = 0을 사용해보자.

10 5x5 행렬을 만들어 체스 보드 패턴으로 채워보자.

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

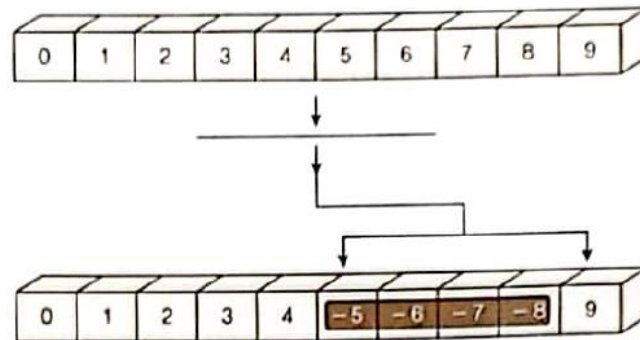
HINT A[1::2, ::2] = 1을 하면 어떻게 될까?

11 3x3 난수로 행렬을 만들고 평균값과 표준 편차로 행렬을 정규화하여 보자.

```
[[-0.93934972 0.57698567 -1.13354651]
 [ 0.17677412 -0.11732708 -0.57457895]
 [ 1.0927037 -1.04232184 1.96066061]]
```

HINT 정규화는 (x - mean)/std을 의미한다.

- 12 넘파이를 사용하여 0에서 9까지의 값을 가진 벡터를 만들고 5에서 8 사이의 숫자 부호를 반전시켜보자.



- 13 넘파이로 3×3 크기의 2차원 배열을 생성하고, 모든 요소의 합, 각 열의 합, 각 행의 합을 계산해보자.

원본 배열:

[[0 1 2]

[3 4 5]

[6 7 8]]

모든 요소의 합: 36

각 열의 합: [9 12 15]

각 행의 합: [3 12 21]

HINT `x = np.arange(0, 9).reshape(3, 3)`을 사용해보자.

- 14 주어진 두 벡터의 내적을 계산하기 위해 넘파이 프로그램을 작성해보자.

원본 벡터 :

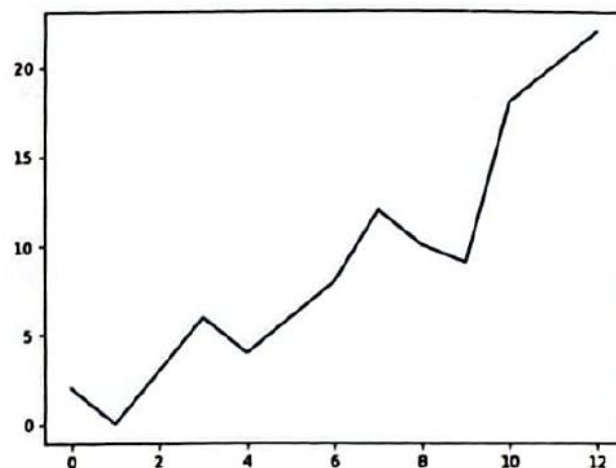
[4 5]

[7 10]

벡터의 내적: 78

HINT @ 연산자나 `dot()` 메소드를 사용한다.

- 15 [2, 0, 3, 6, 4, 6, 8, 12, 10, 9, 18, 20, 22]와 같은 데이터를 이용하여 다음과 같은 선 그래프를 그려보자.



<주의사항>

- 모든 문제의 경우 Python으로 프로그래밍 작업을 수행하고 소스코드와 실행 결과를 모두 캡처하여 답안에 포함시킬 것.
- 본인이름.pdf 파일 1개로 만들어서 과제 게시판에 업로드할 것. (그 외 파일은 미 허용)
- 압축파일 형식으로 업로드하지 말 것.