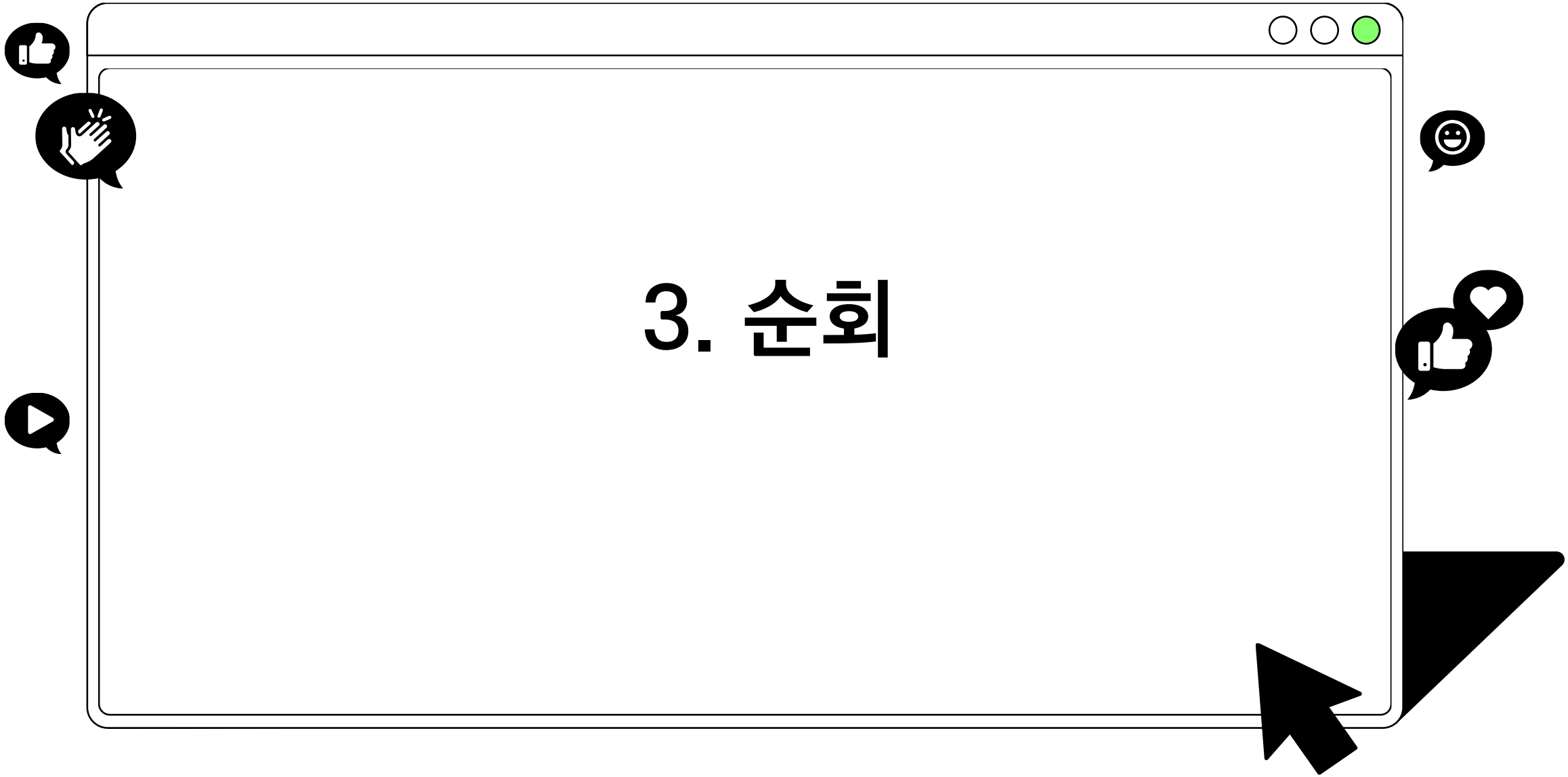
HAPPY HACKING



3. 순회

3. 순회

HAPPY HACKING

이차원 리스트를 단순히 출력하면 아래와 같이 나온다.

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

print(matrix)
>>> [[1, 2, 3, 4], [5, 6, 7, 8], [9, 0, 1, 2]]
```

3. 순회

HAPPY HACKING

이차원 리스트에 담긴 모든 원소를 아래와 같이 출력하고 싶다면 어떻게 할까?

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

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

3. 순회

HAPPY HACKING

인덱스를 통해 각각 출력하면 가능하다 !

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

print(matrix[0][0], matrix[0][1], matrix[0][2], matrix[0][3])
print(matrix[1][0], matrix[1][1], matrix[1][2], matrix[1][3])
print(matrix[2][0], matrix[2][1], matrix[2][2], matrix[2][3])

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

3. 순회

HAPPY HACKING

인덱스를 통해 각각 출력하면 가능하다 !

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

print(matrix[0][0], matrix[0][1], matrix[0][2], matrix[0][3])
print(matrix[1][0], matrix[1][1], matrix[1][2], matrix[1][3])
print(matrix[2][0], matrix[2][1], matrix[2][2], matrix[2][3])

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

하지만 이차원 리스트의 크기가
100 x 100이라도 이렇게 출력할 수 있을까?

3. 순회

HAPPY HACKING

인덱스를 통해 각각 출력하면 가능하다 !

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

print(matrix[0][0], matrix[0][1], matrix[0][2], matrix[0][3])
print(matrix[1][0], matrix[1][1], matrix[1][2], matrix[1][3])
print(matrix[2][0], matrix[2][1], matrix[2][2], matrix[2][3])

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

따라서 이중 반복문을 통해 순회하며
이차원 리스트를 출력한다.

HAPPY HACKING

1. 이중 for문을 이용한 행 우선 순회

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

for i in range(3):
    for j in range(4):
        print(matrix[i][j], end=" ")
    print()

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

HAPPY HACKING

1. 이중 for문을 이용한 행 우선 순회

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

for i in range(3):  # 행
    for j in range(4):  # 열
        print(matrix[i][j], end=" ")
    print()

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

행(i)	열(j)	matrix[i][j]
0	0	1
	1	2
	2	3
	3	4
1	0	5
	1	6
	2	7
	3	8
2	0	9
	1	0
	2	1
	3	2

HAPPY HACKING

1. 이중 for문을 이용한 행 우선 순회

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

for i in range(3):  # 행
    for j in range(4):  # 열
        print(matrix[i][j], end=" ")
    print()
```

하나의 행을 출력

```
>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

행(i)	열(j)	matrix[i][j]
0	0	1
	1	2
	2	3
	3	4
1	0	5
	1	6
	2	7
	3	8
2	0	9
	1	0
	2	1
	3	2

HAPPY HACKING

2. 이중 for문을 이용한 열 우선 순회

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

for i in range(4):
    for j in range(3):
        print(matrix[j][i], end=" ")
    print()

>>> 1 5 9
>>> 2 6 0
>>> 3 7 1
>>> 4 8 2
```

HAPPY HACKING

2. 이중 for문을 이용한 열 우선 순회

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

for i in range(4):  # 열
    for j in range(3):  # 행
        print(matrix[j][i], end=" ")
    print()

>>> 1 5 9
>>> 2 6 0
>>> 3 7 1
>>> 4 8 2
```

열(i)	행(j)	matrix[j][i]
0	0	1
	1	5
	2	9
1	0	2
	1	6
	2	0
2	0	3
	1	7
	2	1
3	0	4
	1	8
	2	2

HAPPY HACKING

2. 이중 for문을 이용한 열 우선 순회

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

for i in range(4):  # 열
    for j in range(3):  # 행
        print(matrix[j][i], end=" ")
    print()
```

하나의 열을 출력

```
>>> 1 5 9
>>> 2 6 0
>>> 3 7 1
>>> 4 8 2
```

열(i)	행(j)	matrix[j][i]
0	0	1
	1	5
	2	9
1	0	2
	1	6
	2	0
2	0	3
	1	7
	2	1
3	0	4
	1	8
	2	2

3. 순회

HAPPY HACKING

행 우선 순회 vs 열 우선 순회

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

for i in range(3):
    for j in range(4):
        print(matrix[i][j], end=" ")
    print()

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

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

for i in range(4):
    for j in range(3):
        print(matrix[j][i], end=" ")
    print()

>>> 1 5 9
>>> 2 6 0
>>> 3 7 1
>>> 4 8 2
```

3. 순회

HAPPY HACKING

행 우선 순회 vs 열 우선 순회

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

for i in range(3):
    for j in range(4):
        print(matrix[i][j], end=" ")
    print()

>>> 1 2 3 4
>>> 5 6 7 8
>>> 9 0 1 2
```

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

for i in range(4):
    for j in range(3):
        print(matrix[j][i], end=" ")
    print()

>>> 1 5 9
>>> 2 6 0
>>> 3 7 1
>>> 4 8 2
```

HAPPY HACKING

행 우선 순회를 이용해 이차원 리스트의 총합 구하기

```
matrix = [
    [1, 1, 1, 1],
    [1, 1, 1, 1],
    [1, 1, 1, 1]
]

total = 0

for i in range(3):
    for j in range(4):
        total += matrix[i][j]

print(total)
>>> 12
```

HAPPY HACKING

[참고] Pythonic한 방법으로 이차원 리스트의 총합 구하기

```
matrix = [
    [1, 1, 1, 1],
    [1, 1, 1, 1],
    [1, 1, 1, 1]
]

total = sum(map(sum, matrix))

print(total)
>>> 12
```

3. 순회

HAPPY HACKING

행 우선 순회를 이용해 이차원 리스트의 최대값, 최소값 구하기

```
# 최대값
matrix = [
    [0, 5, 3, 1],
    [4, 6, 10, 8],
    [9, -1, 1, 5]
]

max_value = 0

for i in range(3):
    for j in range(4):
        if matrix[i][j] > max_value:
            max_value = matrix[i][j]

print(max_value)
>>> 10
```

```
# 최소값
matrix = [
    [0, 5, 3, 1],
    [4, 6, 10, 8],
    [9, -1, 1, 5]
]

min_value = 99999999

for i in range(3):
    for j in range(4):
        if matrix[i][j] < min_value:
            min_value = matrix[i][j]

print(min_value)
>>> -1
```

HAPPY HACKING

[참고] Pythonic한 방법으로 이차원 리스트의 최대값, 최소값 구하기

```
matrix = [
    [0, 5, 3, 1],
    [4, 6, 10, 8],
    [9, -1, 1, 5]
]

max_value = max(map(max, matrix))
min_value = min(map(min, matrix))

print(max_value)
>>> 10

print(min_value)
>>> -1
```

3. 순회

HAPPY HACKING

이차원 리스트 순회 연습

문제

목록

2행 3열 리스트 두 개에 각각의 값을 입력 받은 후 두 배열의 같은 위치끼리 곱하여 새로운 리스트에
저장한 후 출력하는 프로그램을 작성하시오.

입력 예

복사하기

```
(출력)first array
3 6 9
8 5 2
(출력)second array
9 8 7
6 5 4
```

출력 예

복사하기

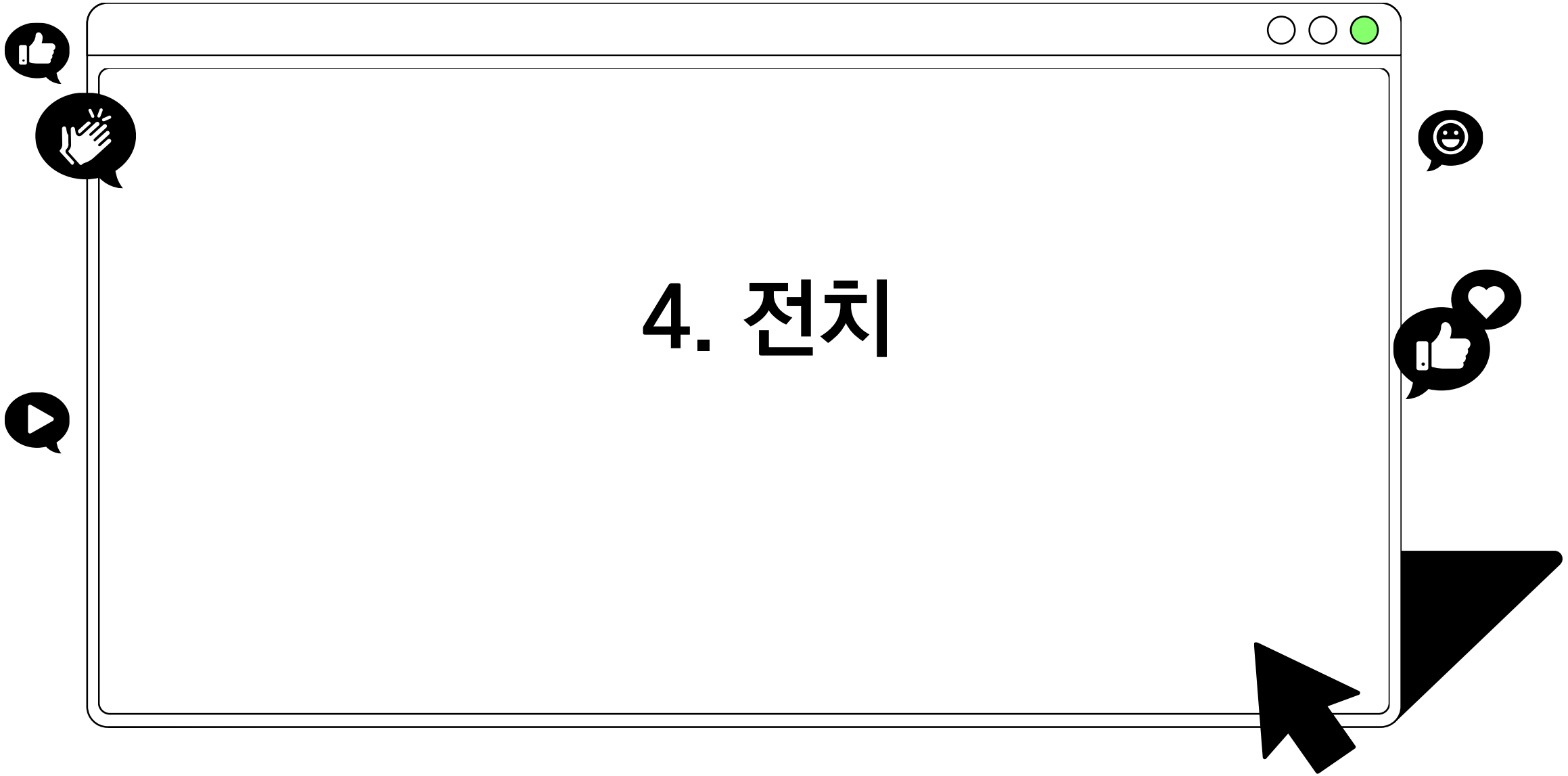
```
27 48 63
48 25 8
```

http://jungol.co.kr/bbs/board.php?bo_table=pbank&wr_id=4364&sca=pyc0

HAPPY HACKING

이차원 리스트 순회 연습 정답

```
list_a = [list(map(int, input().split())) for i in range(2)]
list_b = [list(map(int, input().split())) for i in range(2)]
list_c = [[0] * 3 for _ in range(2)]

# 두 배열의 같은 위치끼리 곱하여 새로운 이차원 리스트에 저장
for i in range(2):
    for j in range(3):
        list_c[i][j] = list_a[i][j] * list_b[i][j]

# 결과 출력
for i in range(2):
    for j in range(3):
        print(list_c[i][j], end=" ")
    print()
```



4. 전치

4. 전치

HAPPY HACKING

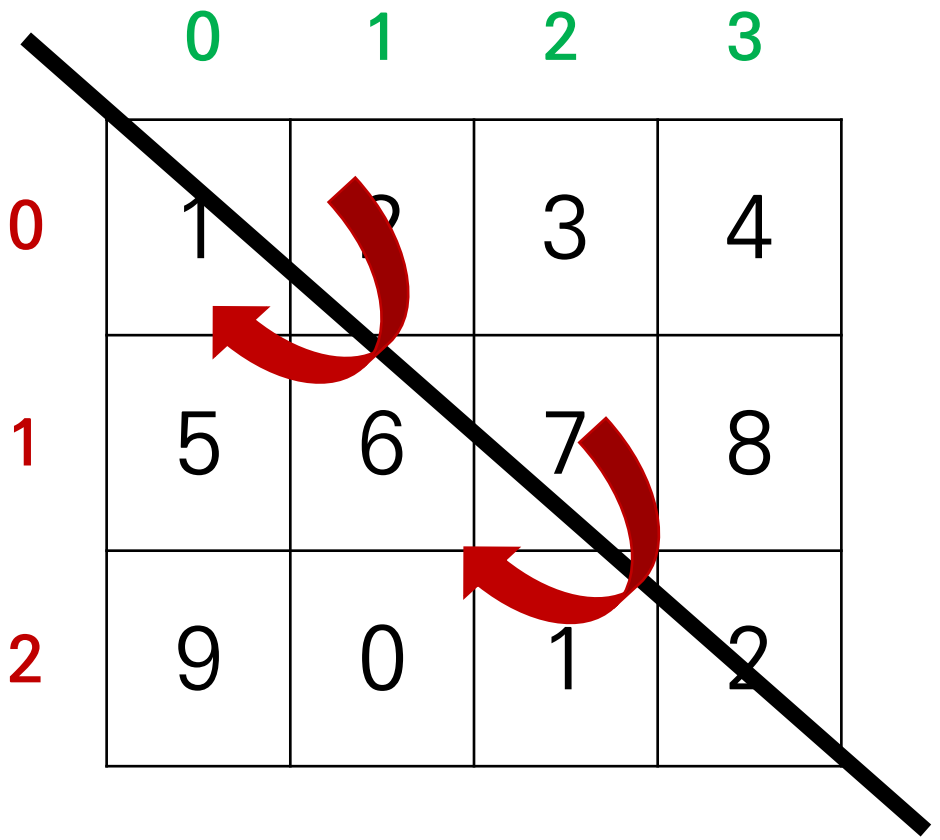
전치(transpose)란 행렬의 행과 열을 서로 맞바꾸는 것을 의미한다.

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	0	1	2

4. 전치

HAPPY HACKING

전치(transpose)란 행렬의 행과 열을 서로 맞바꾸는 것을 의미한다.



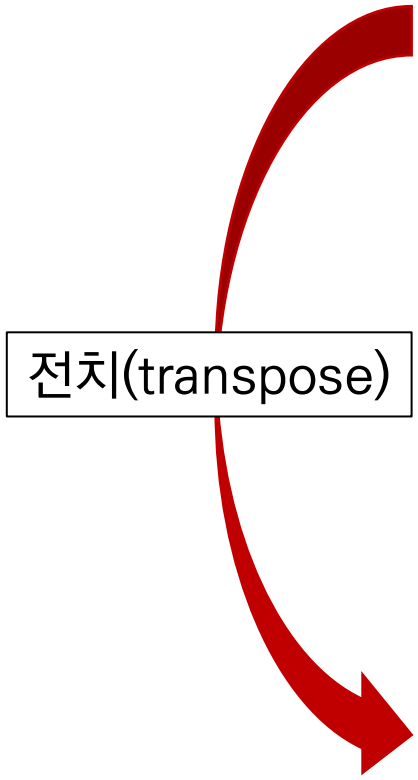
4. 전치

HAPPY HACKING

전치(transpose)란 행렬의 행과 열을 서로 맞바꾸는 것을 의미한다.

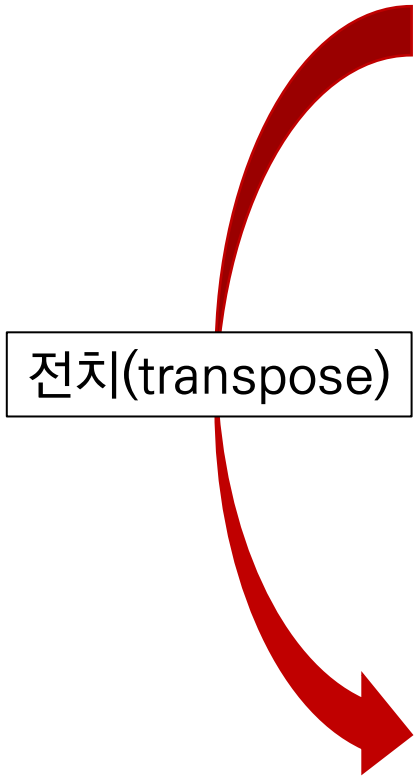
	0	1	2
0	1	5	9
1	2	6	0
2	3	7	1
3	4	8	2

4. 전치




```python
matrix = [
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 0, 1, 2]
]

transposed_matrix = [[0] * 3 for _ in range(4)]

for i in range(4):
    for j in range(3):
        transposed_matrix[i][j] = matrix[j][i]  # 행, 열 맞바꾸기

"""
transposed_matrix = [
    [1, 5, 9],
    [2, 6, 0],
    [3, 7, 1],
    [4, 8, 2]
]
"""
```

4. 전치

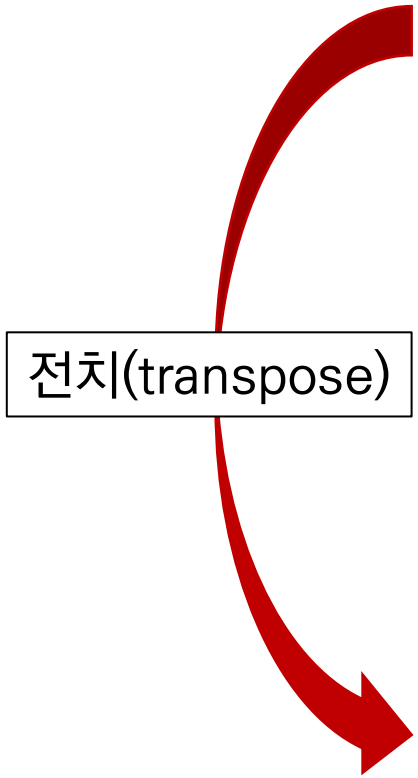



```python
matrix = [
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 0, 1, 2]
]

transposed_matrix = [[0] * 3 for _ in range(4)]

for i in range(4):
    for j in range(3):
        transposed_matrix[i][j] = matrix[j][i]  # 행, 열 맞바꾸기

"""
transposed_matrix = [
    [1, 5, 9],
    [2, 6, 0],
    [3, 7, 1],
    [4, 8, 2]
]
"""
```

전치 행렬을 담을 이차원 리스트 초기화
(행과 열의 크기가 반대)

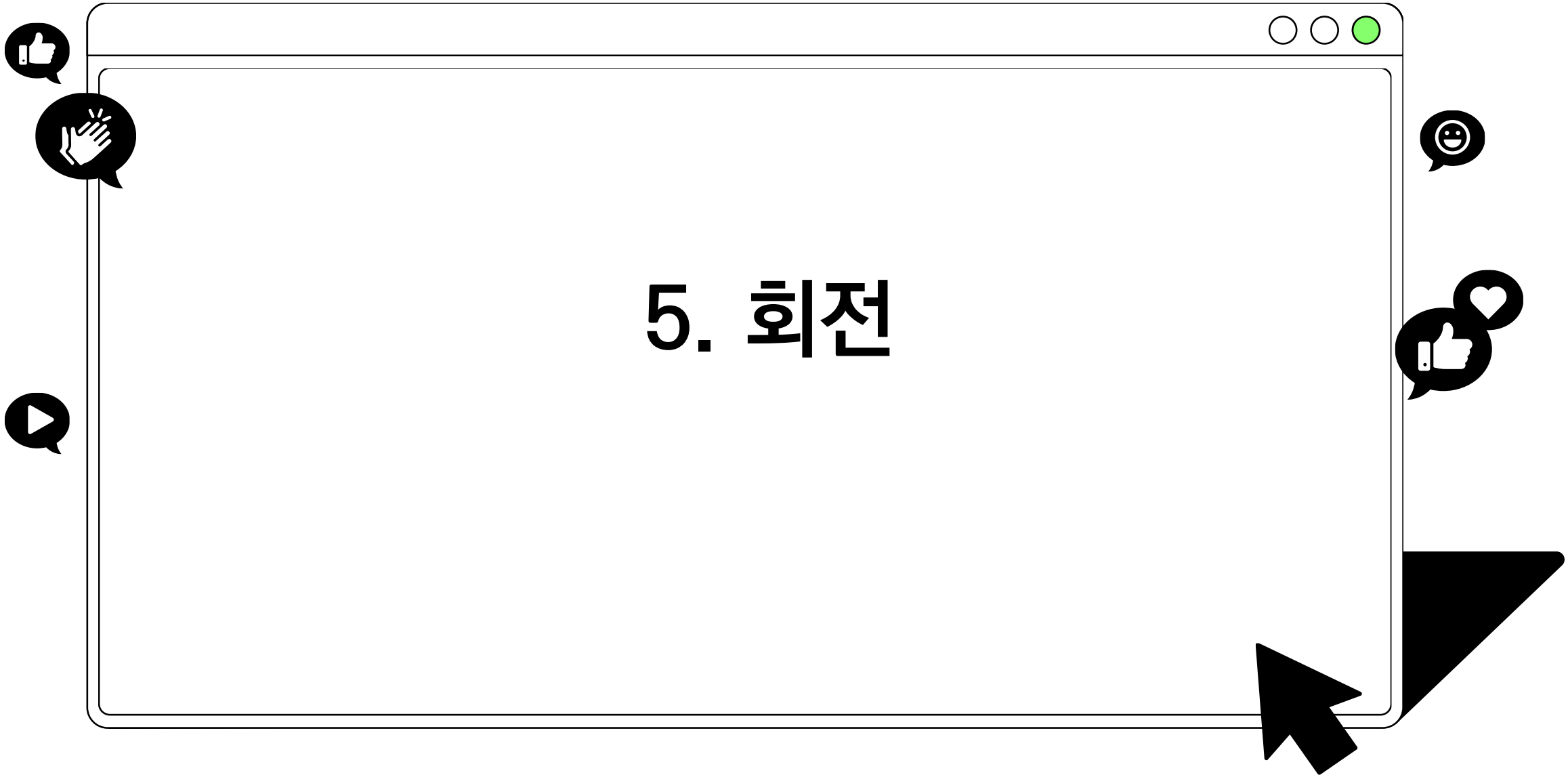
4. 전치

HAPPY HACKING




```python
matrix = [
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 0, 1, 2]
]

transposed_matrix = [[0] * 3 for _ in range(4)]

for i in range(4):
    for j in range(3):
        transposed_matrix[i][j] = matrix[j][i]  # 행, 열 맞바꾸기

"""
transposed_matrix = [
    [1, 5, 9],
    [2, 6, 0],
    [3, 7, 1],
    [4, 8, 2]
]
"""
```

HAPPY HACKING

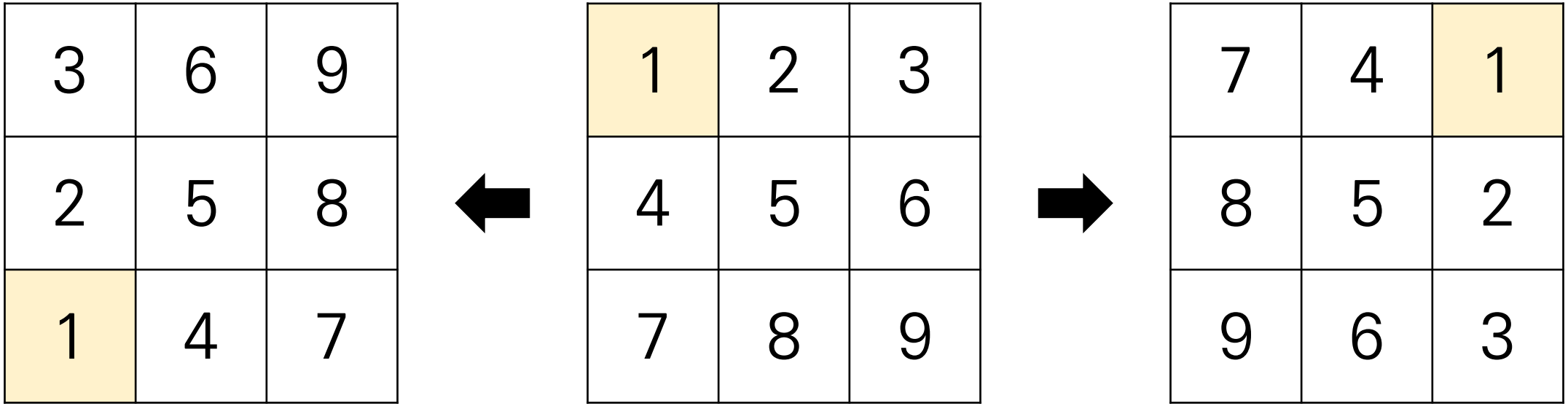


5. 회전

5. 회전

HAPPY HACKING

문제에서 이차원 리스트를 왼쪽, 오른쪽으로 90도 회전하는 경우가 존재한다.



왼쪽 90도 회전 | 원본 | 오른쪽 90도 회전

5. 회전

HAPPY HACKING

1. 왼쪽으로 90도 회전하기

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

n = 3
rotated_matrix = [[0] * n for _ in range(n)]

for i in range(n):
    for j in range(n):
        rotated_matrix[i][j] = matrix[j][n-i-1]
```

2. 오른쪽으로 90도 회전하기

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

n = 3
rotated_matrix = [[0] * n for _ in range(n)]

for i in range(n):
    for j in range(n):
        rotated_matrix[i][j] = matrix[n-j-1][i]
```

5. 회전

HAPPY HACKING

1. 왼쪽으로 90도 회전하기

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

n = 3
rotated_matrix = [[0] * n for _ in range(n)]

for i in range(n):
    for j in range(n):
        rotated_matrix[i][j] = matrix[j][n-i-1]
```

```
"""
rotated_matrix = [
    [3, 6, 9],
    [2, 5, 8],
    [1, 4, 7]
]
"""
```

2. 오른쪽으로 90도 회전하기

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

n = 3
rotated_matrix = [[0] * n for _ in range(n)]

for i in range(n):
    for j in range(n):
        rotated_matrix[i][j] = matrix[n-j-1][i]
```

```
"""
rotated_matrix = [
    [7, 4, 1],
    [8, 5, 2],
    [9, 6, 3]
]
"""
```

직접 작성해보면서 왼쪽, 오른쪽 회전이 각각 어떻게 동작하는지 파악해보자 !