

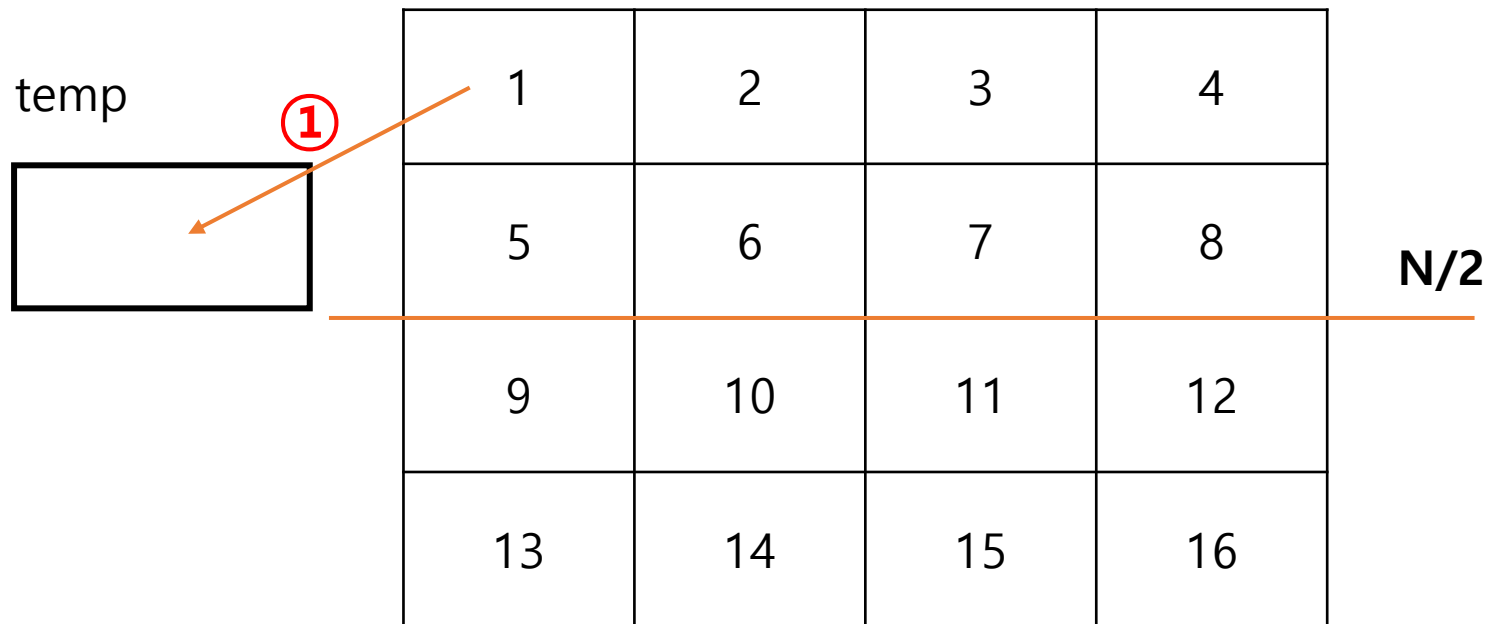
# Algorithm

---

- 배열돌리기 3

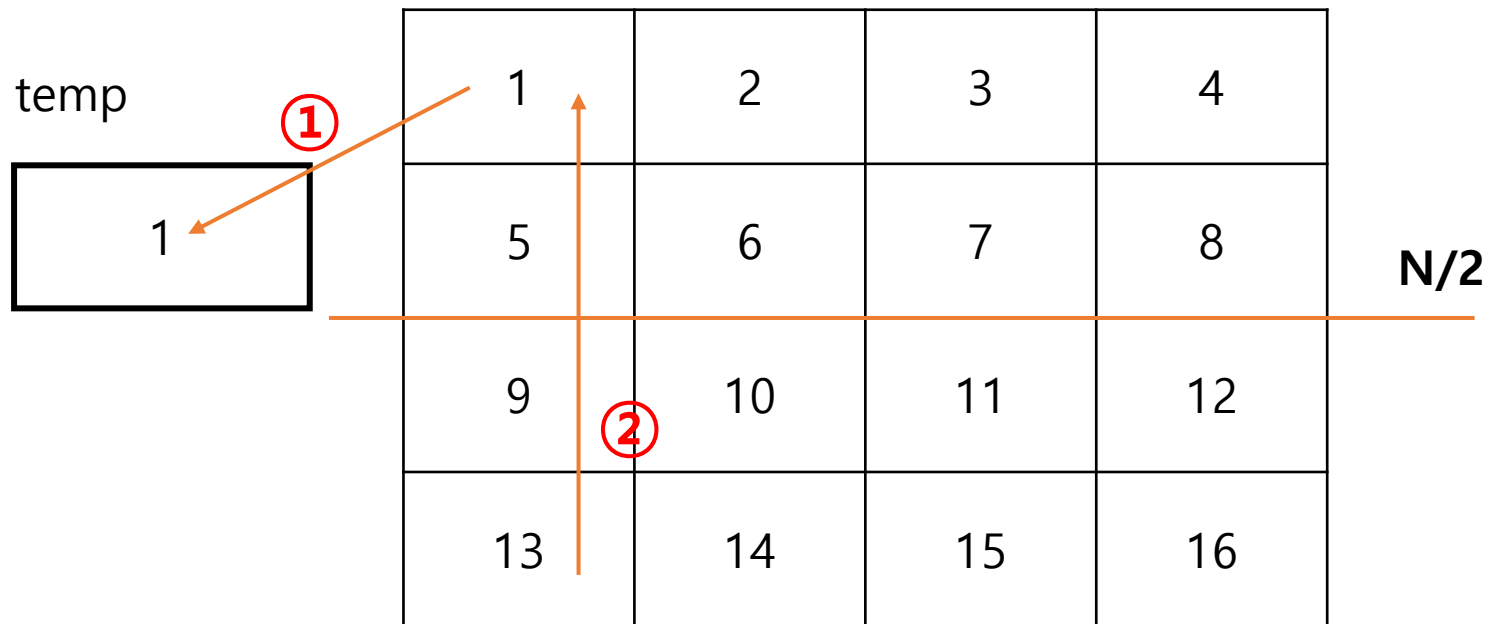
# 16935-배열돌리기3

## 1. 상하 반전



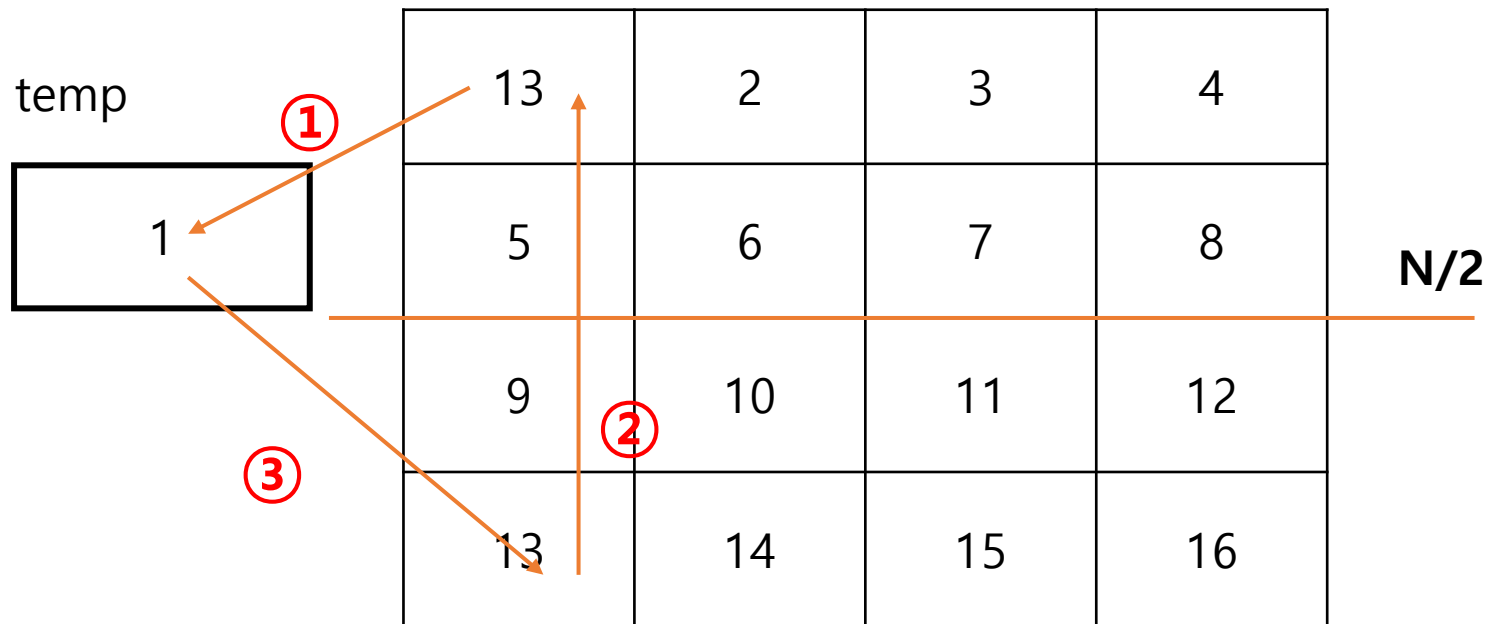
# 16935-배열돌리기3

## 1. 상하 반전



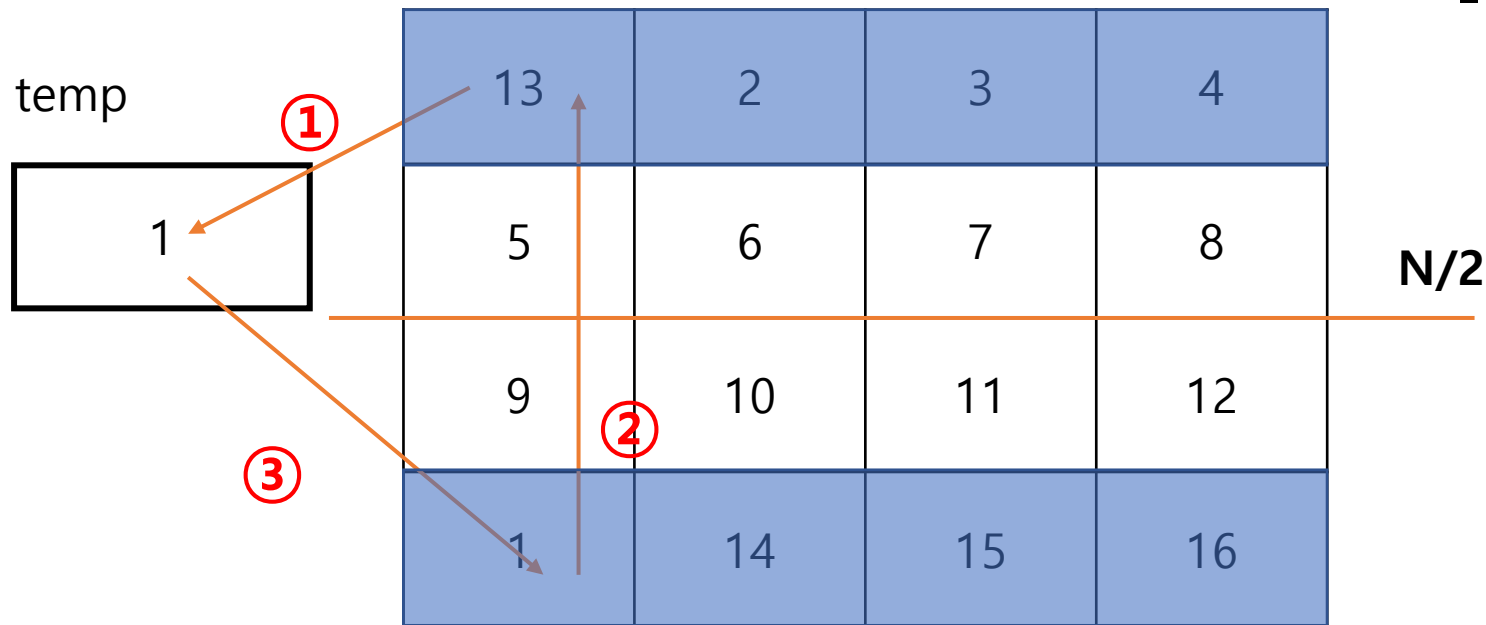
# 16935-배열돌리기3

## 1. 상하 반전



# 16935-배열돌리기3

## 1. 상하 반전



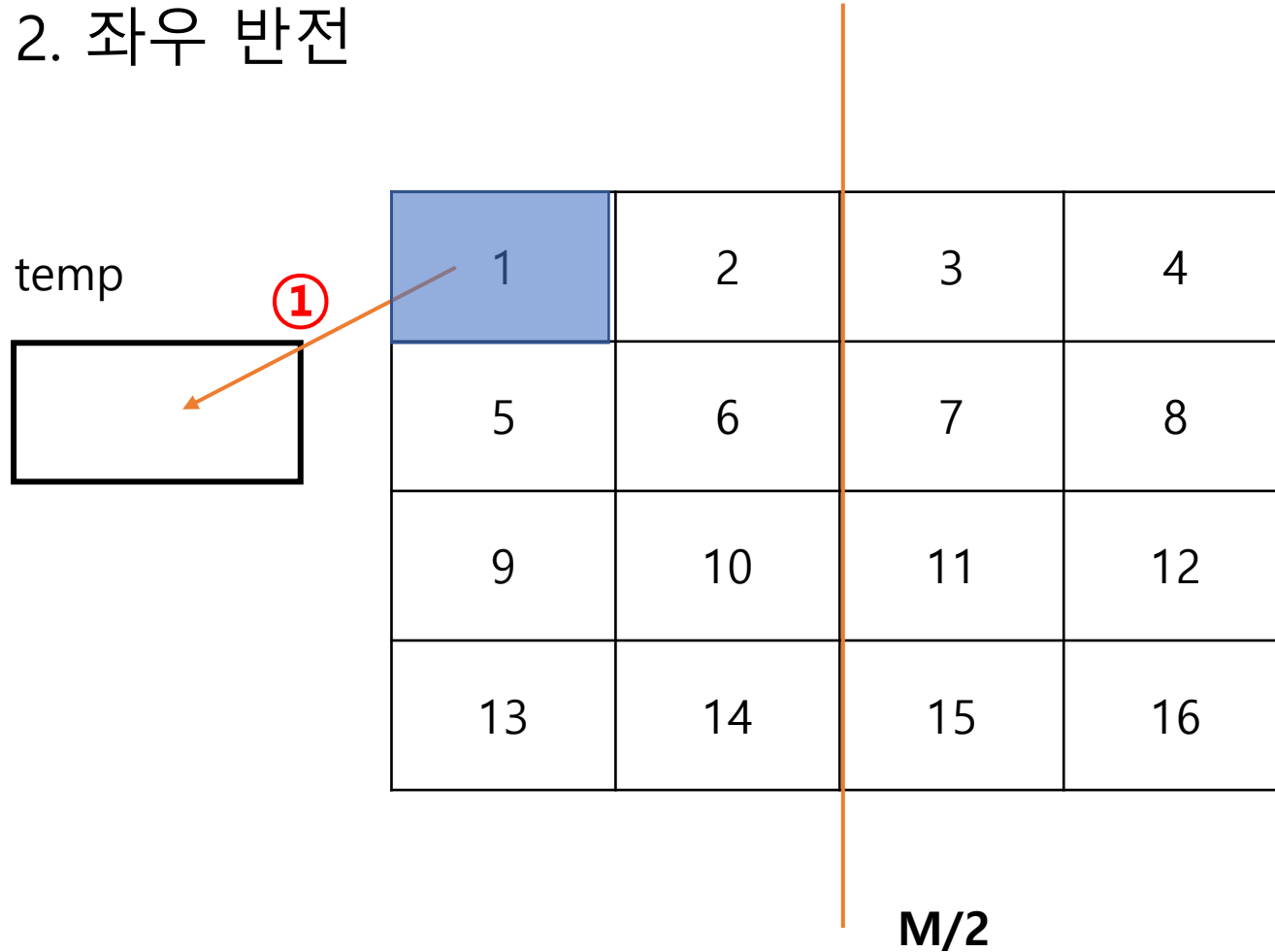
- 1차원 배열 한번에 바꿔 줘도 됨.

- 주소값이 바뀌는 것

```
for(i=0;i<N/2;i++){  
    int[] tmp = map[i];  
    map[i]=map[N-i-1];  
    map[N-i-1]=tmp;  
}
```

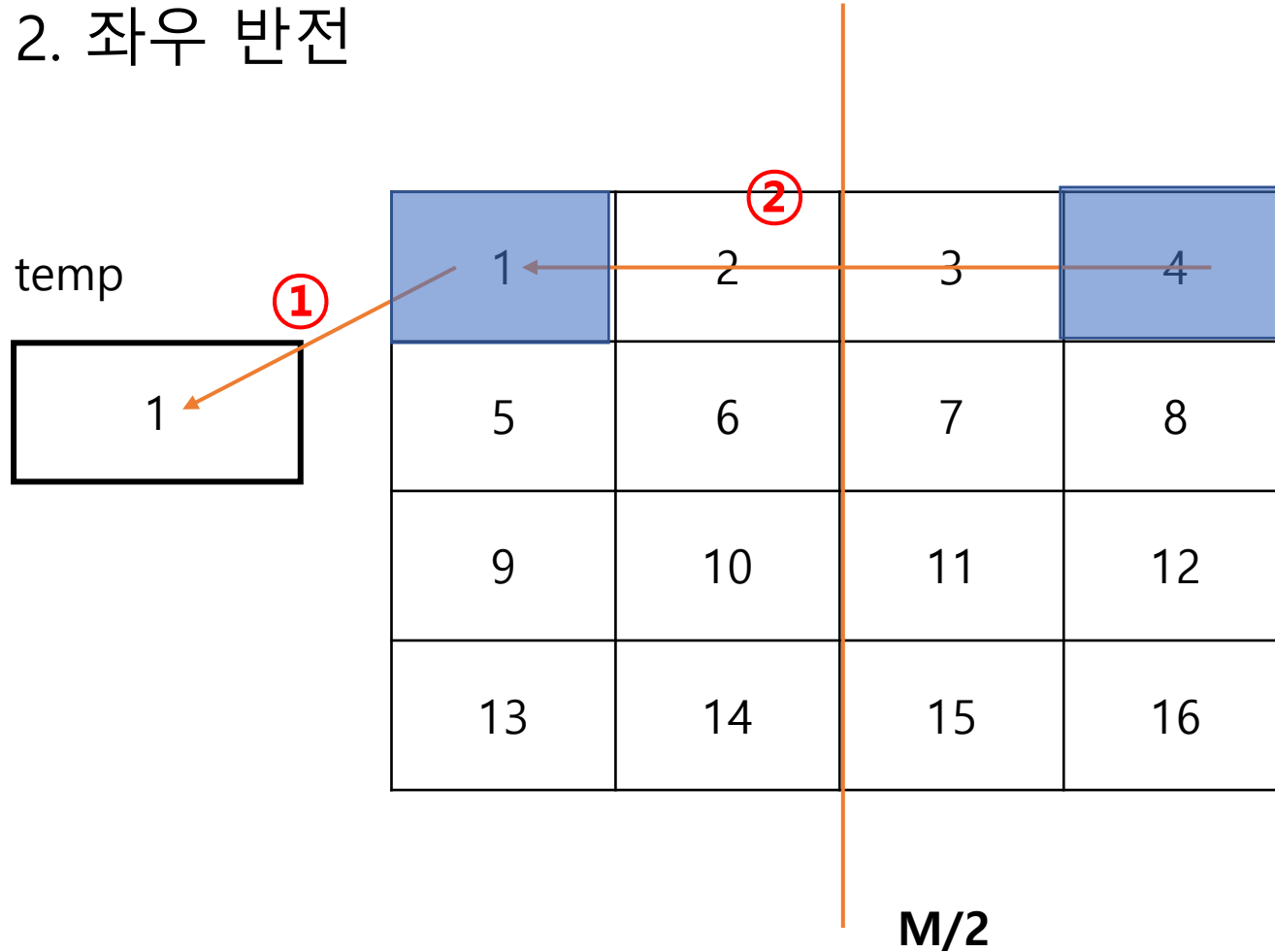
# 16935-배열돌리기3

## 2. 좌우 반전



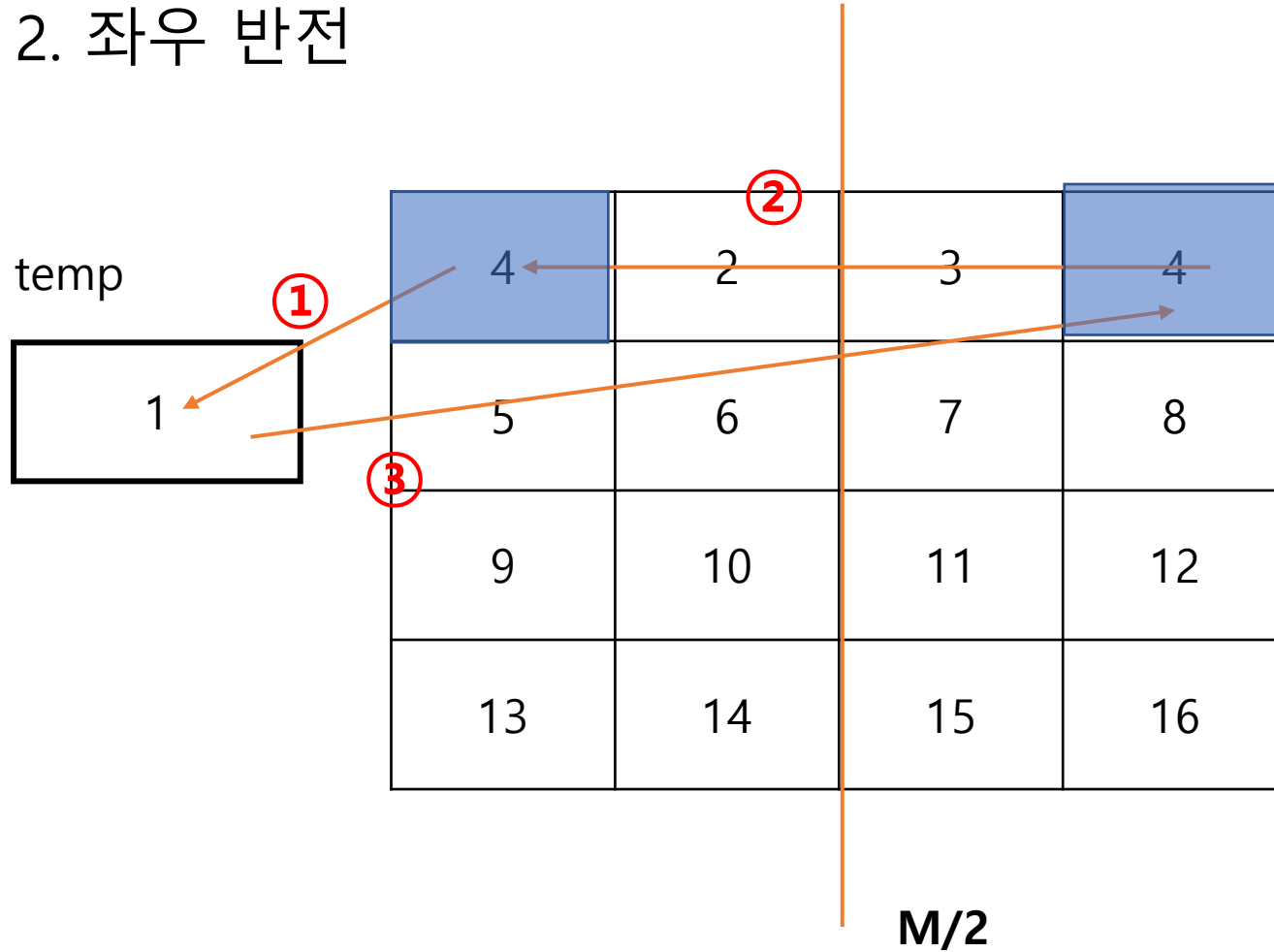
# 16935-배열돌리기3

## 2. 좌우 반전



# 16935-배열돌리기3

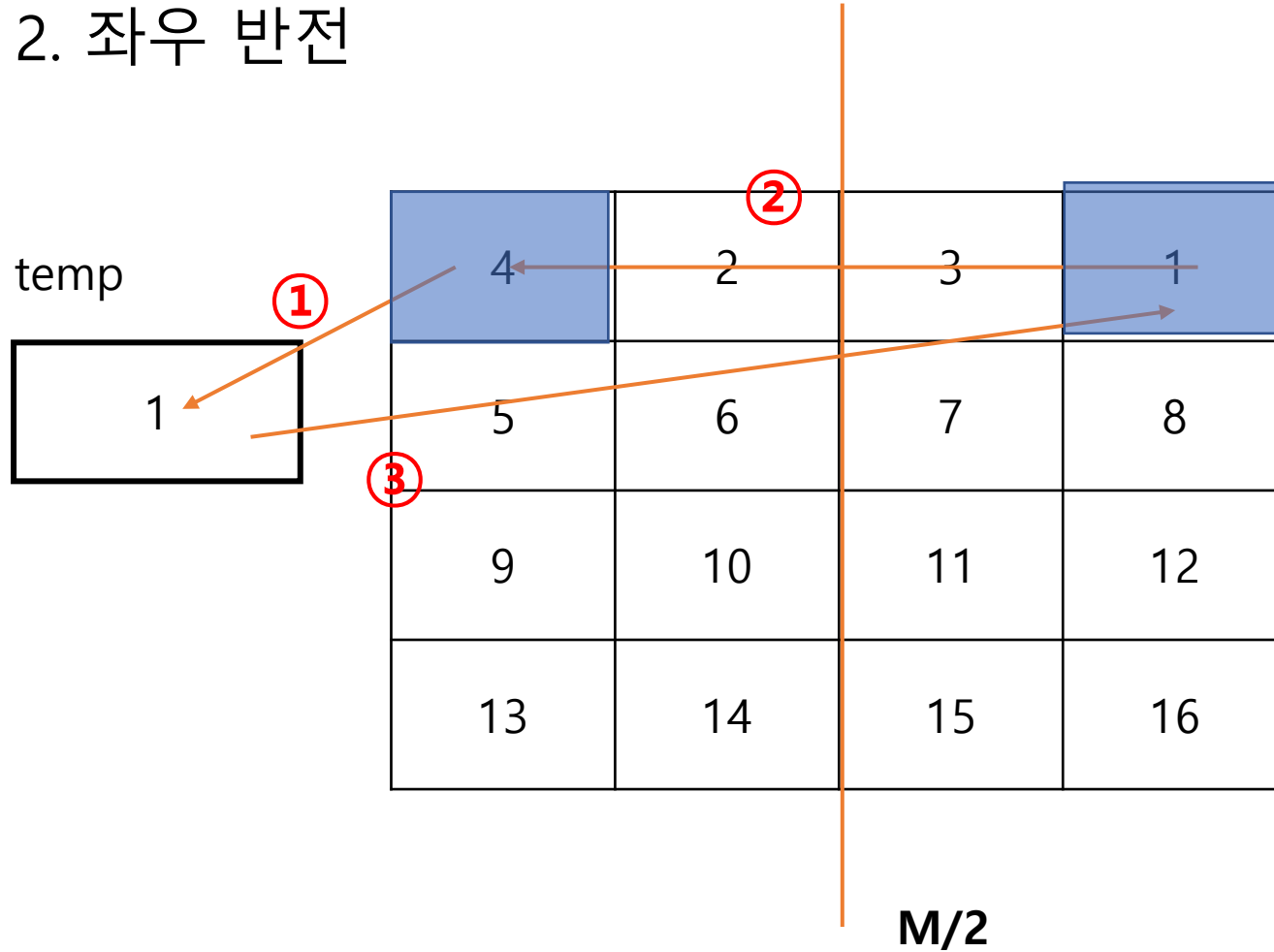
## 2. 좌우 반전





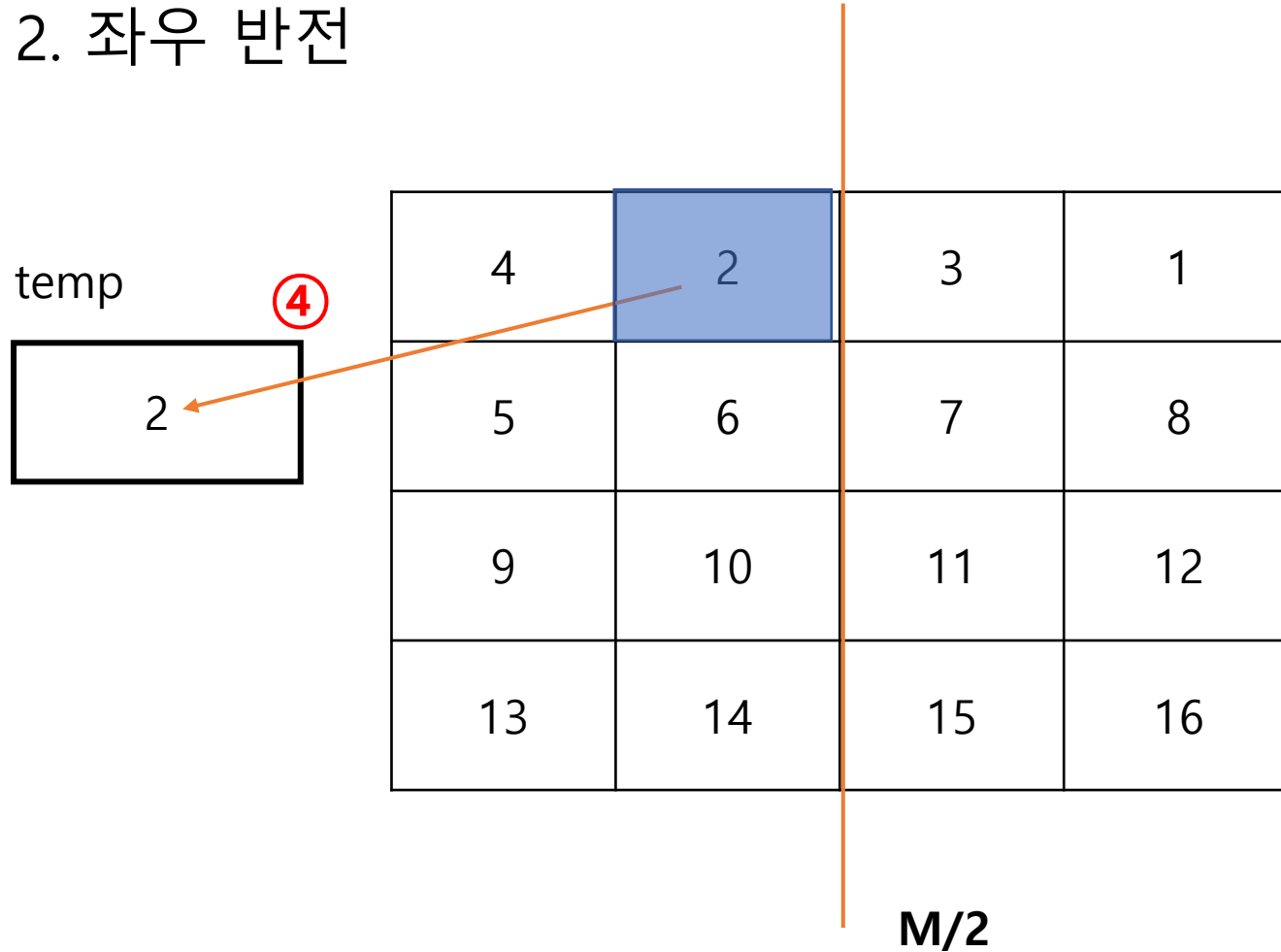
# 16935-배열돌리기3

## 2. 좌우 반전



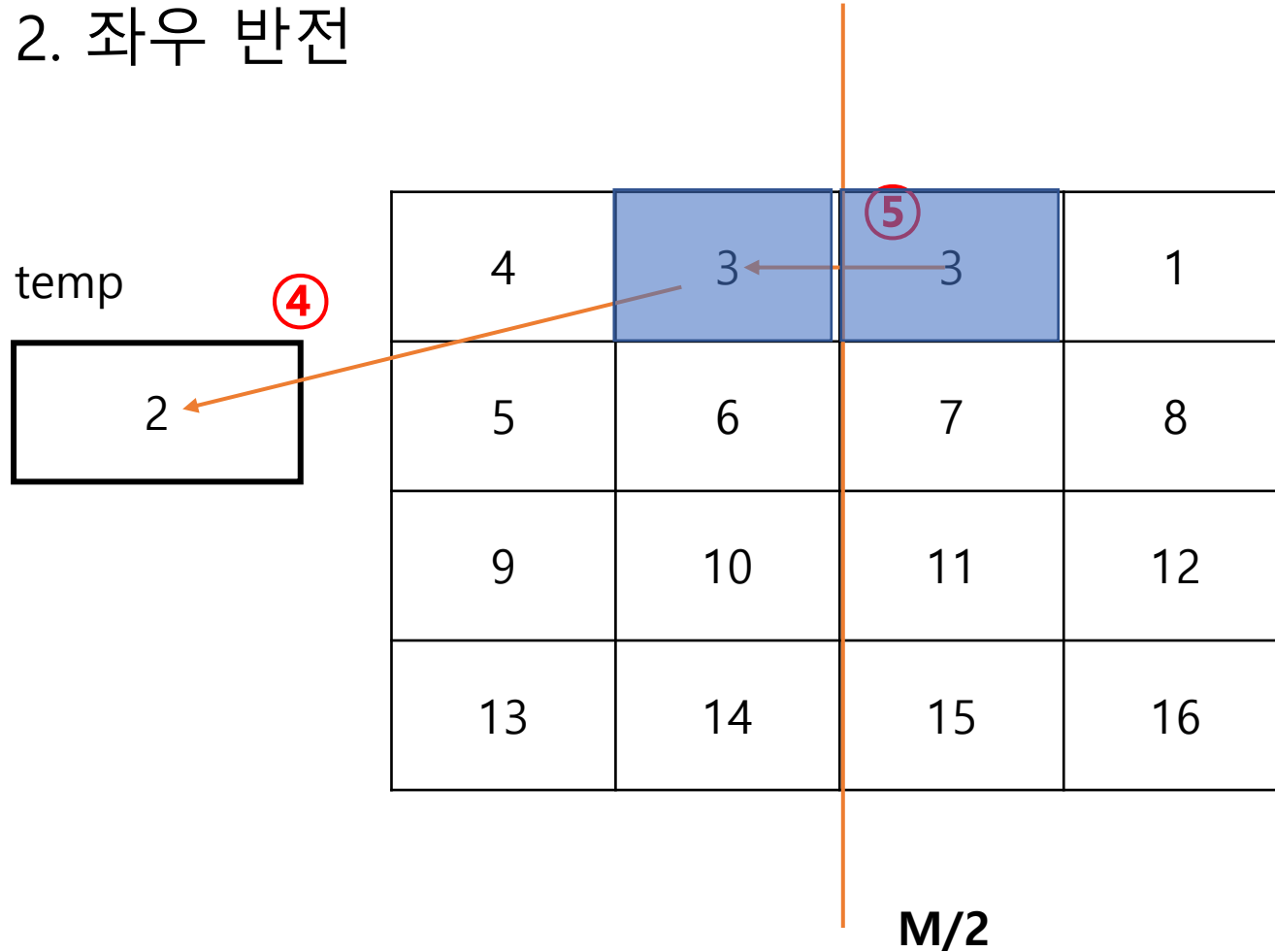
# 16935-배열돌리기3

## 2. 좌우 반전



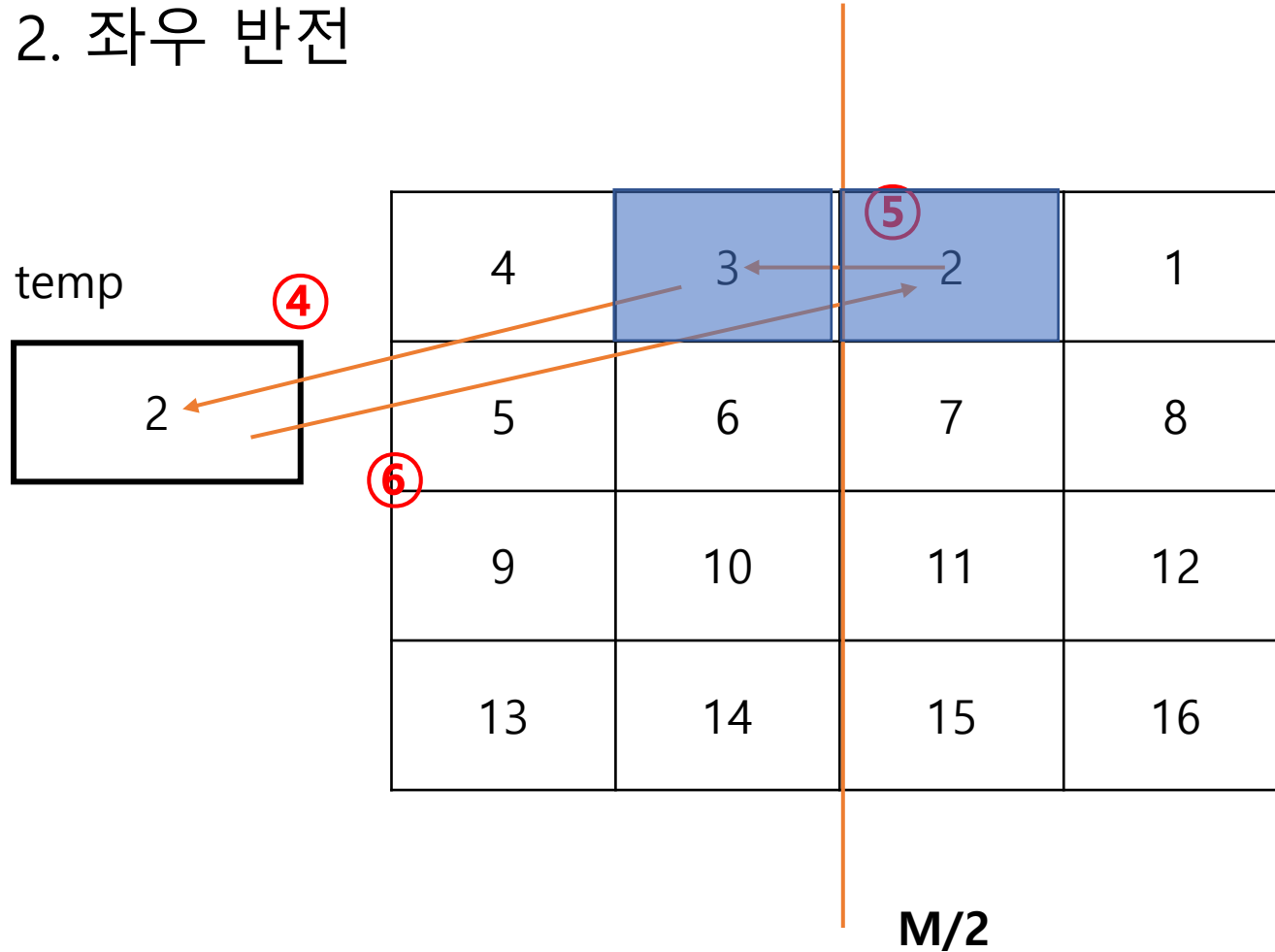
# 16935-배열돌리기3

## 2. 좌우 반전



# 16935-배열돌리기3

## 2. 좌우 반전



- 상하 반전과 달리 1개씩 값을 바꿔줘야함

```
for(i=0;i<N;i++){  
    for(j=0;j<M/2;j++){  
        int temp = map[i][j];  
        map[i][j]=map[i][M-j-1];  
        map[i][M-j-1]=temp;  
    }  
}
```

# 16935-배열돌리기3

## 3. 오른쪽으로 90도 회전(배열 새로 생성)

map[0,0]

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

tmp[0,3]

			1

# 16935-배열돌리기3

## 3. 오른쪽으로 90도 회전(배열 새로 생성)

map[0,1]

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

tmp[1,3]

			1
			2

map[i][j] ---> tmp[j][N-i-1] 이동

# 16935-배열돌리기3

## 3. 오른쪽으로 90도 회전(배열 새로 생성)

map[0,1]

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



tmp[1,3]

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

map = tmp --> 배열 주소값 넘겨주기  
혹은 원본 배열로 하드카피!!

# 16935-배열돌리기3

3. 오른쪽으로 90도 회전(배열 새로 생성)

```
for (int i=0; i<N; i++){  
    for (int j=0; j<M; j++) {  
        tmp[j][N-i-1] = map[i][j];  
    }  
}  
map = tmp;
```

- 
- tmp라는 배열은 호출 할때마다 새로 생성한다는 가정에만 map=tmp 됨.
  - 만일 둘다 static인 경우 같은 주소값을 레퍼런스해서 값이 이상해질 수 있음



# 16935-배열돌리기3

## 4. 왼쪽으로 90도 회전 (배열 새로 생성)

map[0,0]

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

tmp[3,0]

1			

# 16935-배열돌리기3

## 4. 왼쪽으로 90도 회전 (배열 새로 생성)

map[0,1]

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

tmp[2,0]

2			
1			

# 16935-배열돌리기3

## 4. 왼쪽으로 90도 회전 (배열 새로 생성)

map

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

tmp

4	8	12	16
3	7	11	15
2	6	10	14
1	5	9	13

$\text{map}[i][j] \rightarrow \text{tmp}[M-j-1][i]$  이동

# 16935-배열돌리기3

## 4. 왼쪽으로 90도 회전 (배열 새로 생성)

map

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



tmp

4	8	12	16
3	7	11	15
2	6	10	14
1	5	9	13

map = tmp --> 배열 주소값 넘겨주기  
혹은 원본으로 하드카피!!!!

# 16935-배열돌리기3

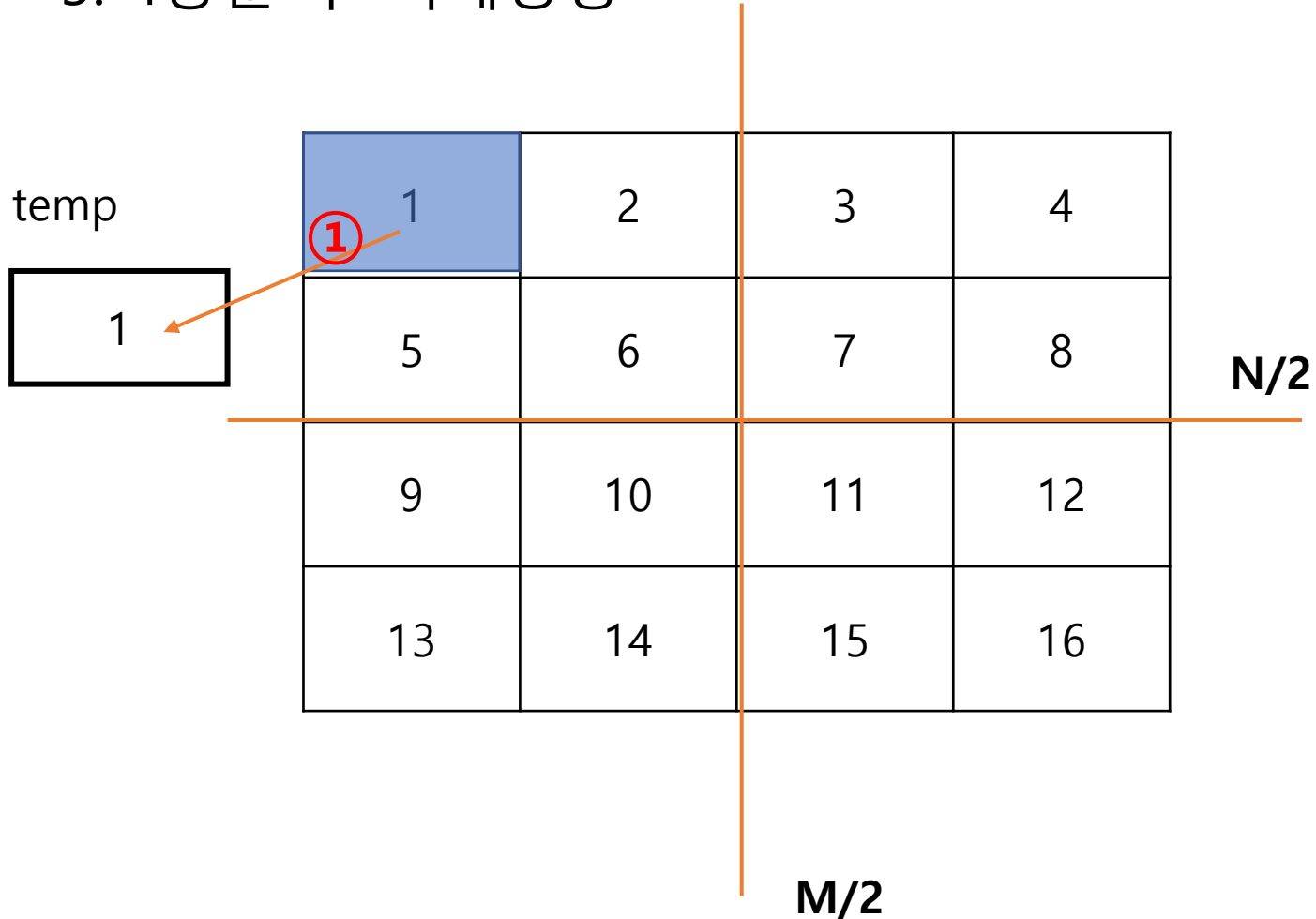
---

## 4. 왼쪽으로 90도 회전 (배열 새로 생성)

```
for (int i=0; i<N; i++) {  
    for (int j=0; j<M; j++){  
        tmp[M-j-1][i] = map[i][j];  
    }  
}  
map = tmp;
```

# 16935-배열돌리기3

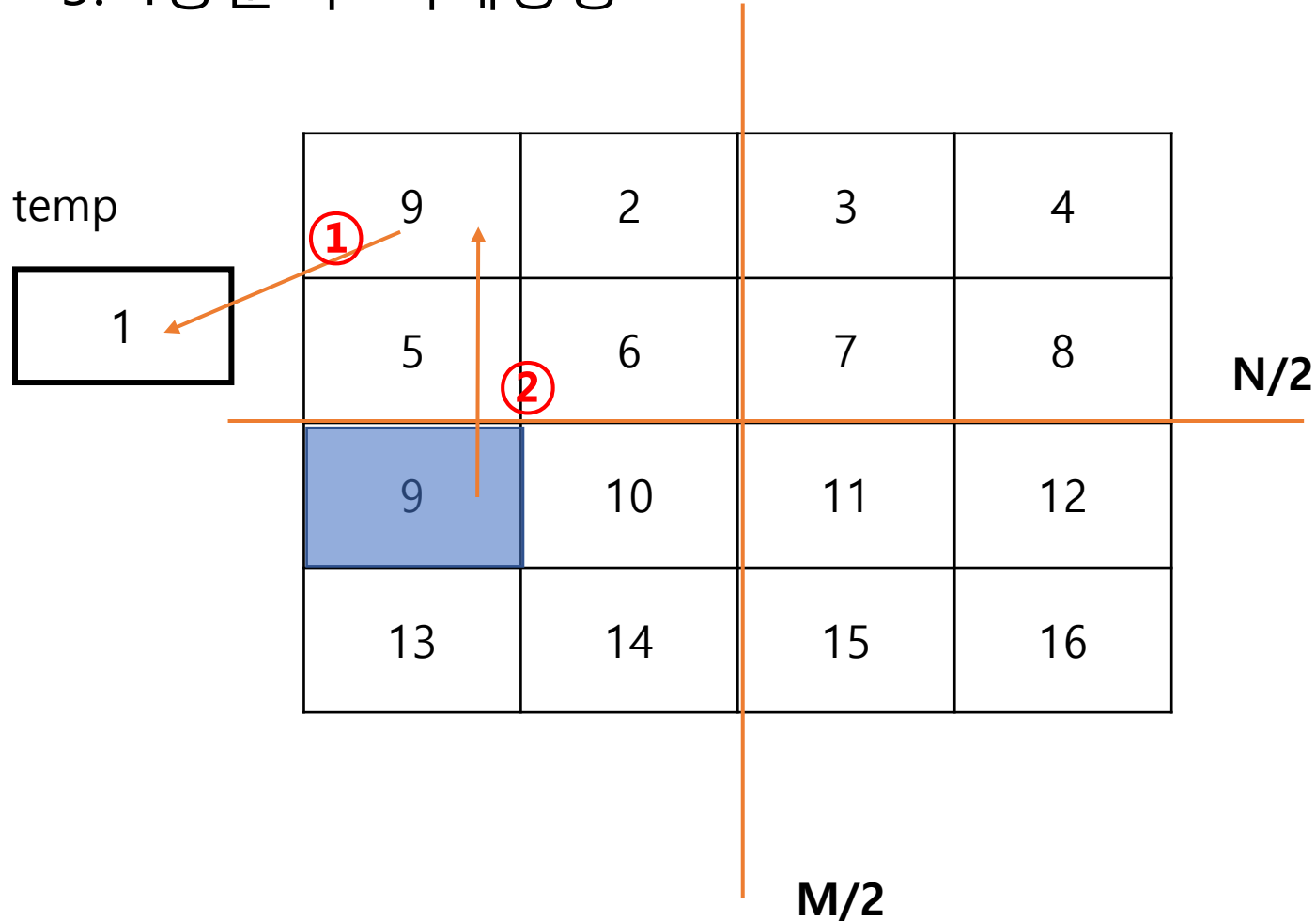
## 5. 4등분 후 시계방향



N/2, M/2 범위 고정

# 16935-배열돌리기3

## 5. 4등분 후 시계방향

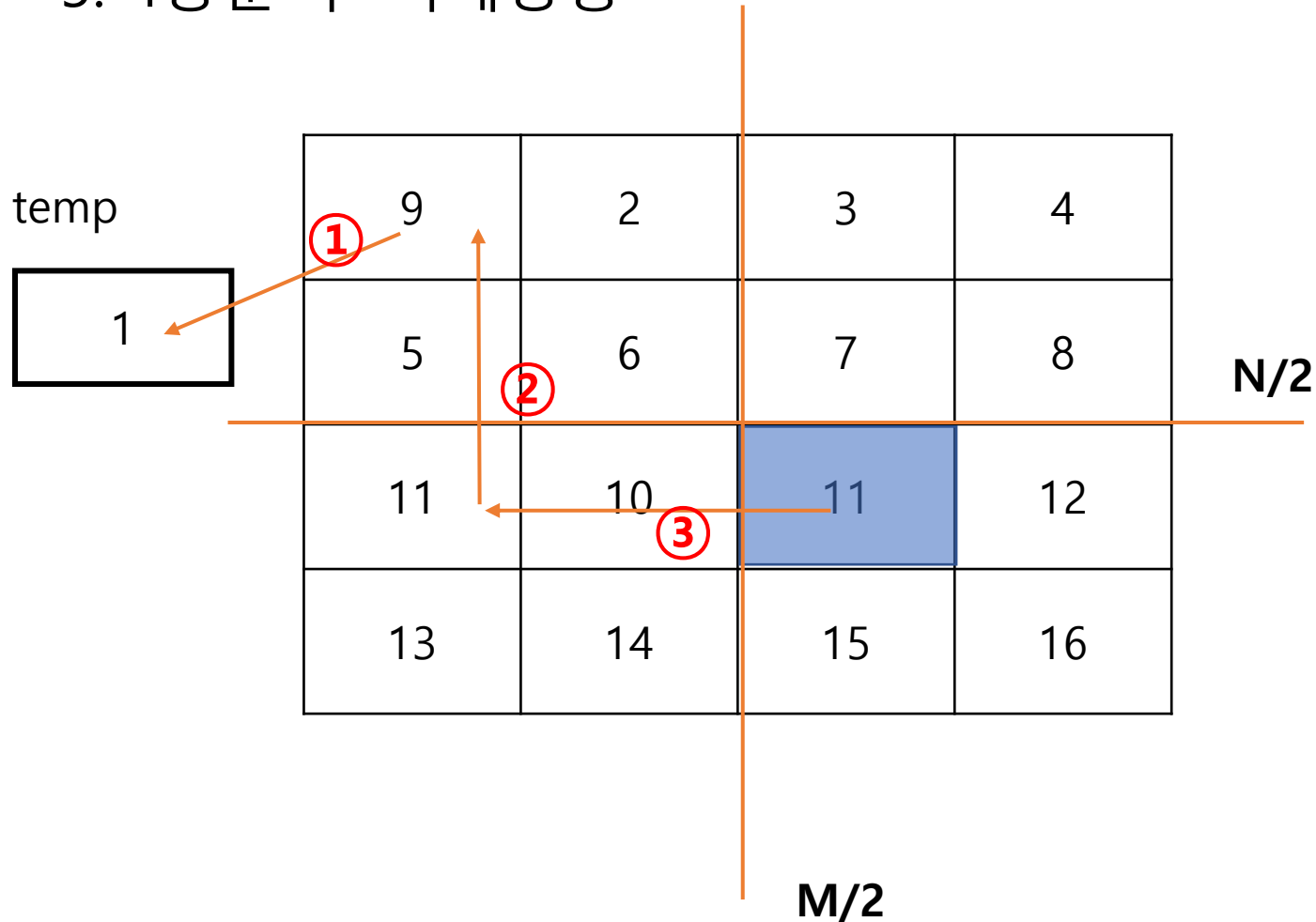


N/2, M/2 범위 고정

- 그룹별로 돌리기

# 16935-배열돌리기3

## 5. 4등분 후 시계방향



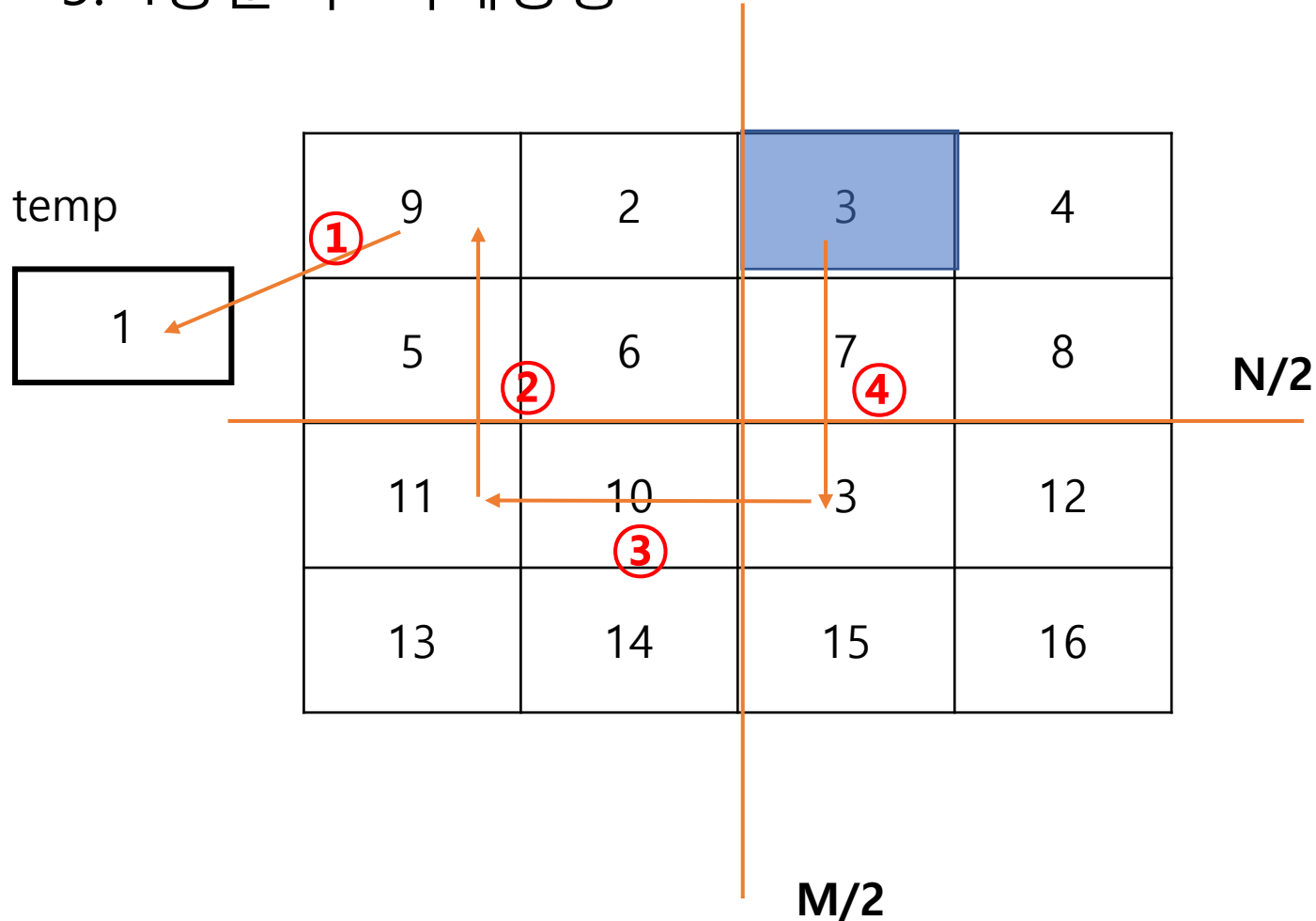
$N/2, M/2$  범위 고정

- 그룹별로 돌리기



# 16935-배열돌리기3

## 5. 4등분 후 시계방향

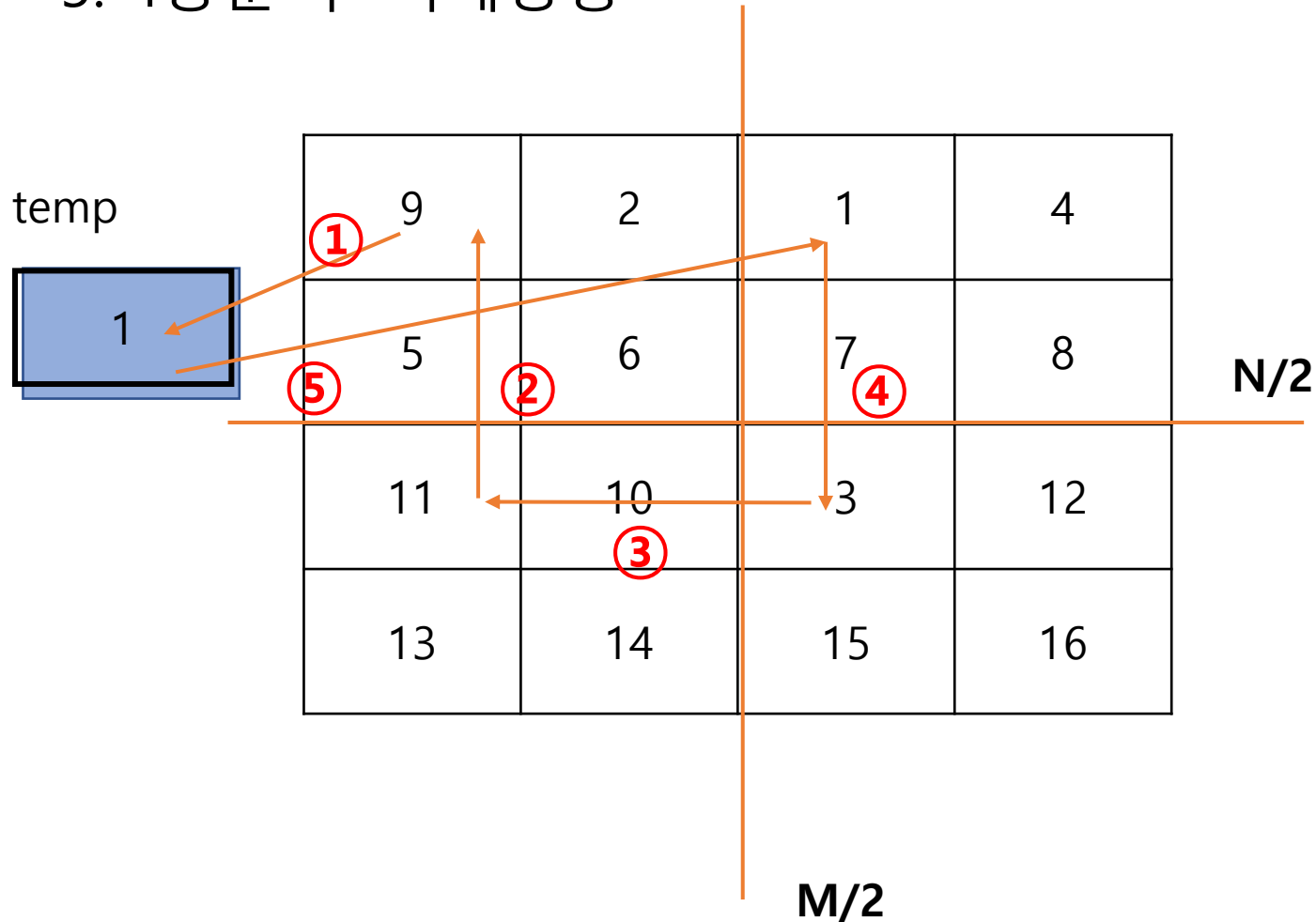


N/2, M/2 범위 고정

- 그룹별로 돌리기

# 16935-배열돌리기3

## 5. 4등분 후 시계방향

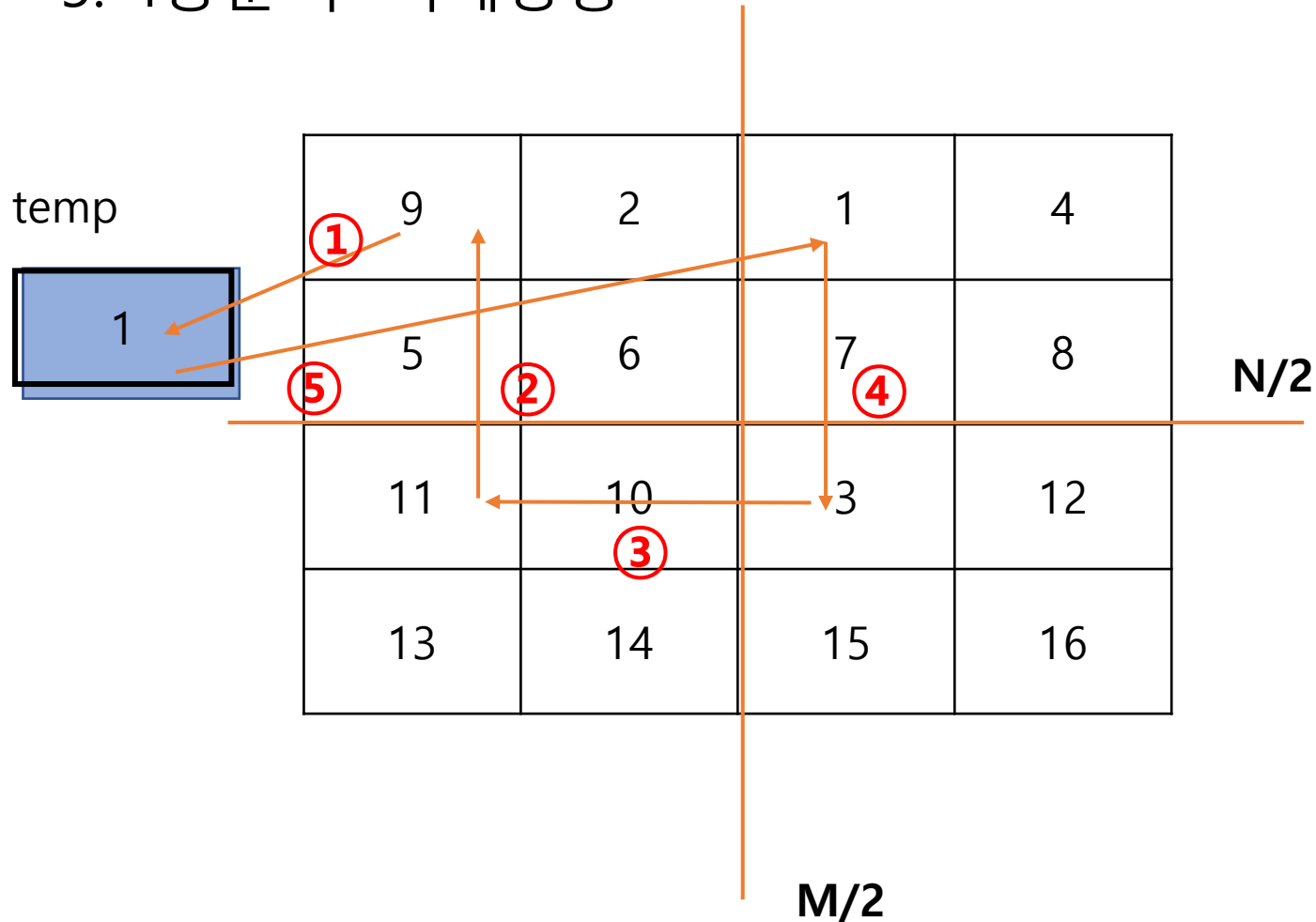


N/2, M/2 범위 고정

- 그룹별로 돌리기

# 16935-배열돌리기3

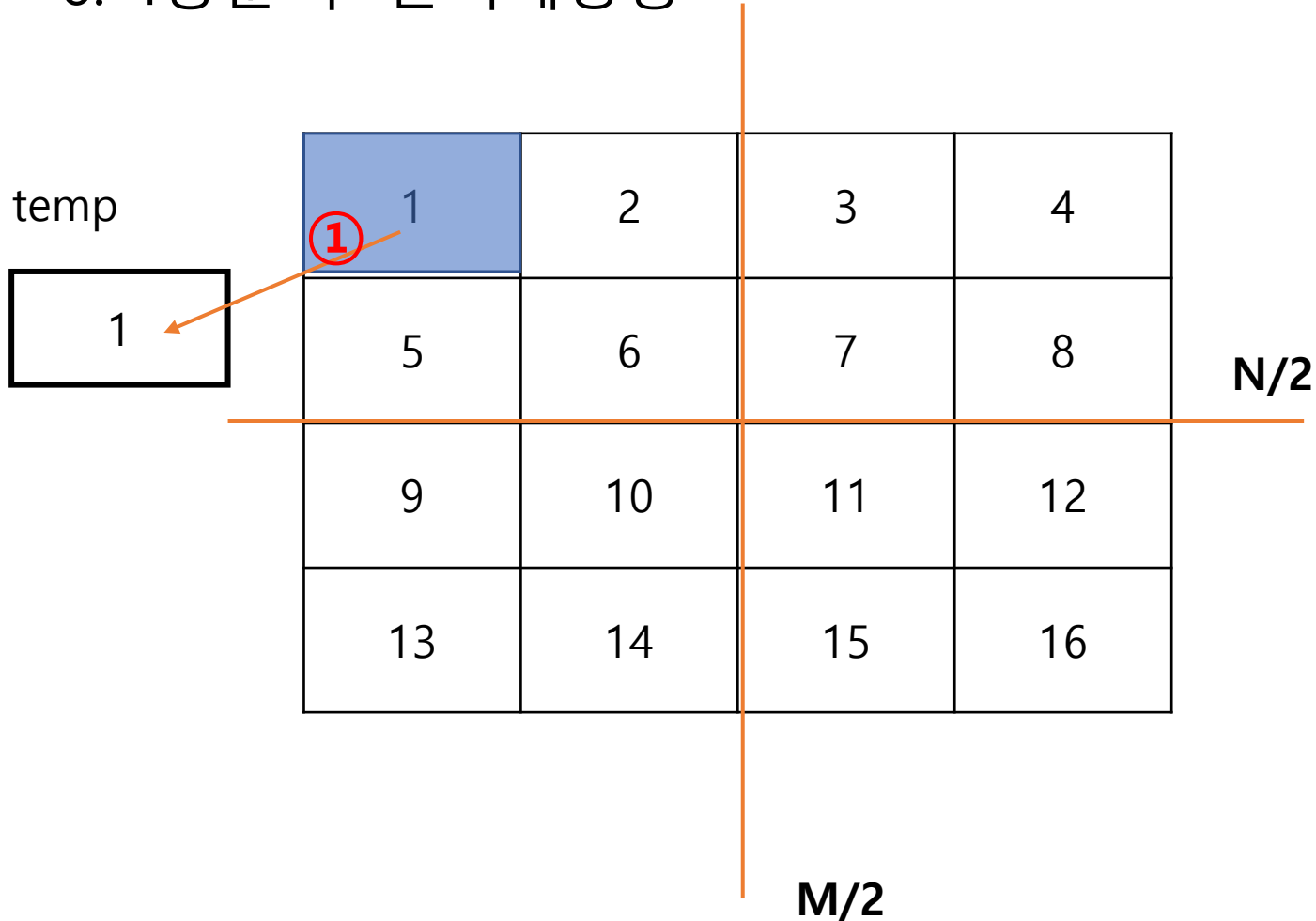
## 5. 4등분 후 시계방향



```
for (int i=0, h=N/2; i<h; i++){  
    for (int j=0, l=M/2; j<l; j++){  
        int tmp = map[i][j];  
        map[i][j] = map[i+h][j];  
        map[i+h][j] = map[i+h][j+l];  
        map[i+h][j+l] = map[i][j+l];  
        map[i][j+l] = tmp;  
    }  
}
```

# 16935-배열돌리기3

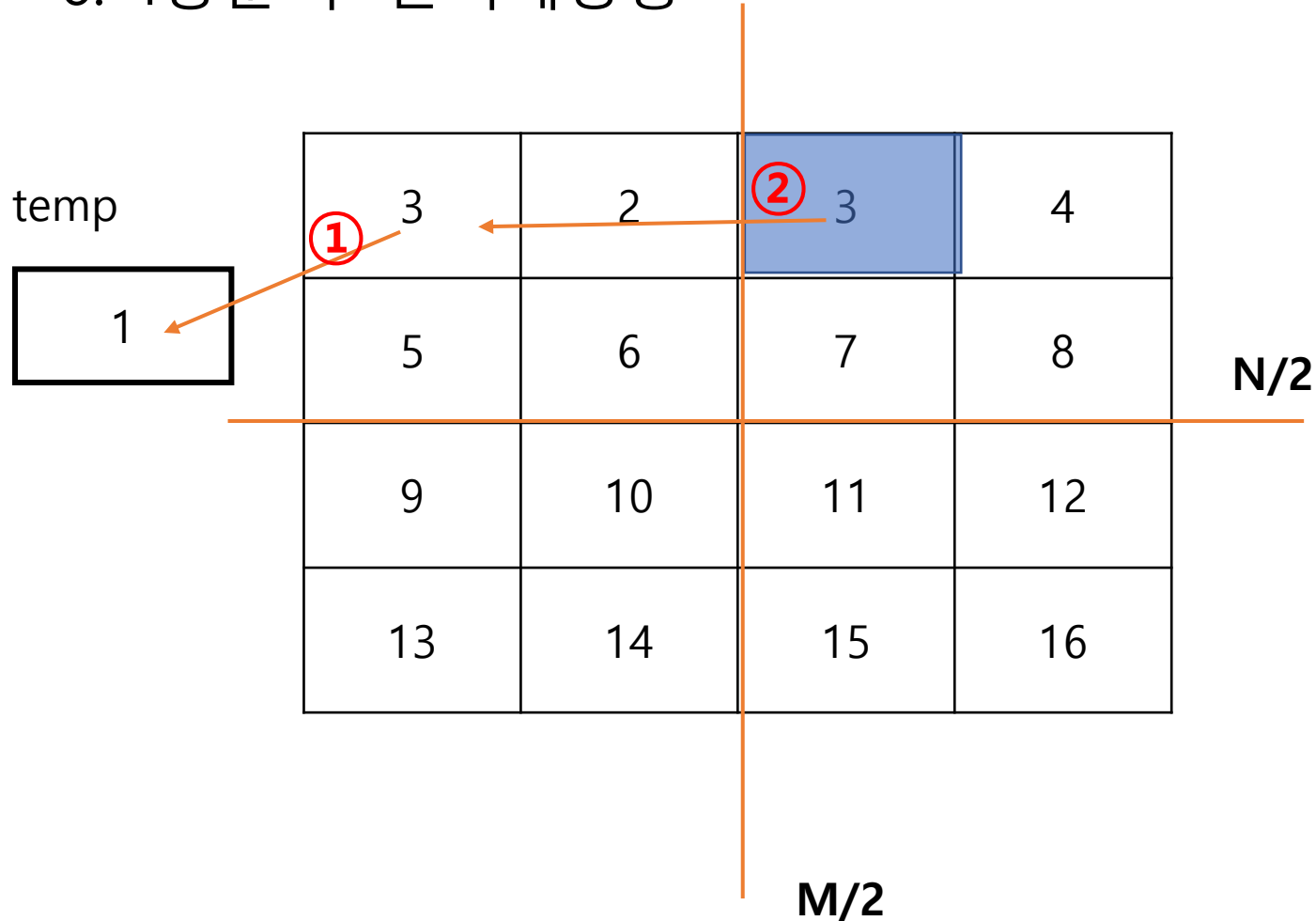
## 6. 4등분 후 반시계방향



N/2, M/2 범위 고정

# 16935-배열돌리기3

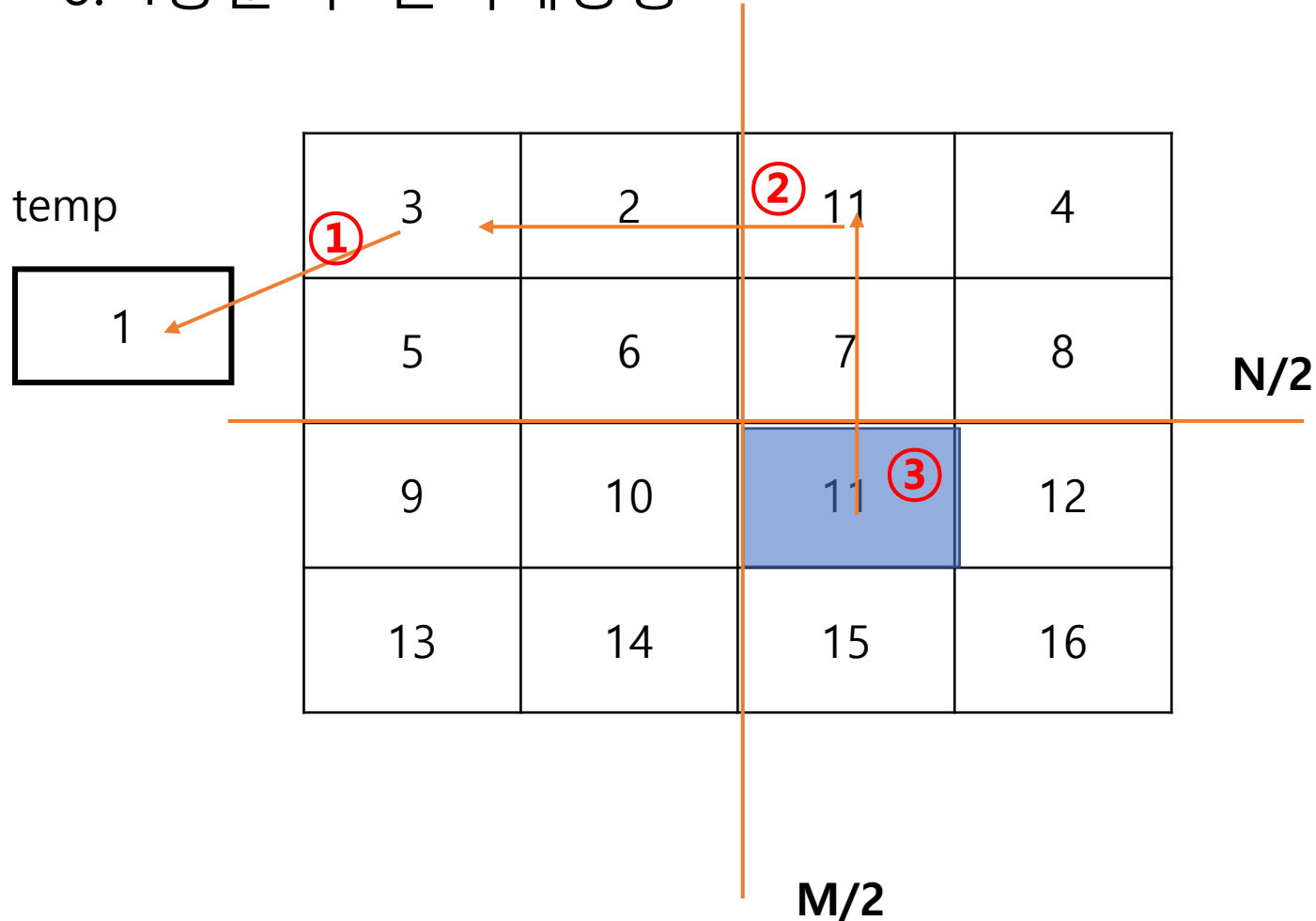
## 6. 4등분 후 반시계방향



N/2, M/2 범위 고정

# 16935-배열돌리기3

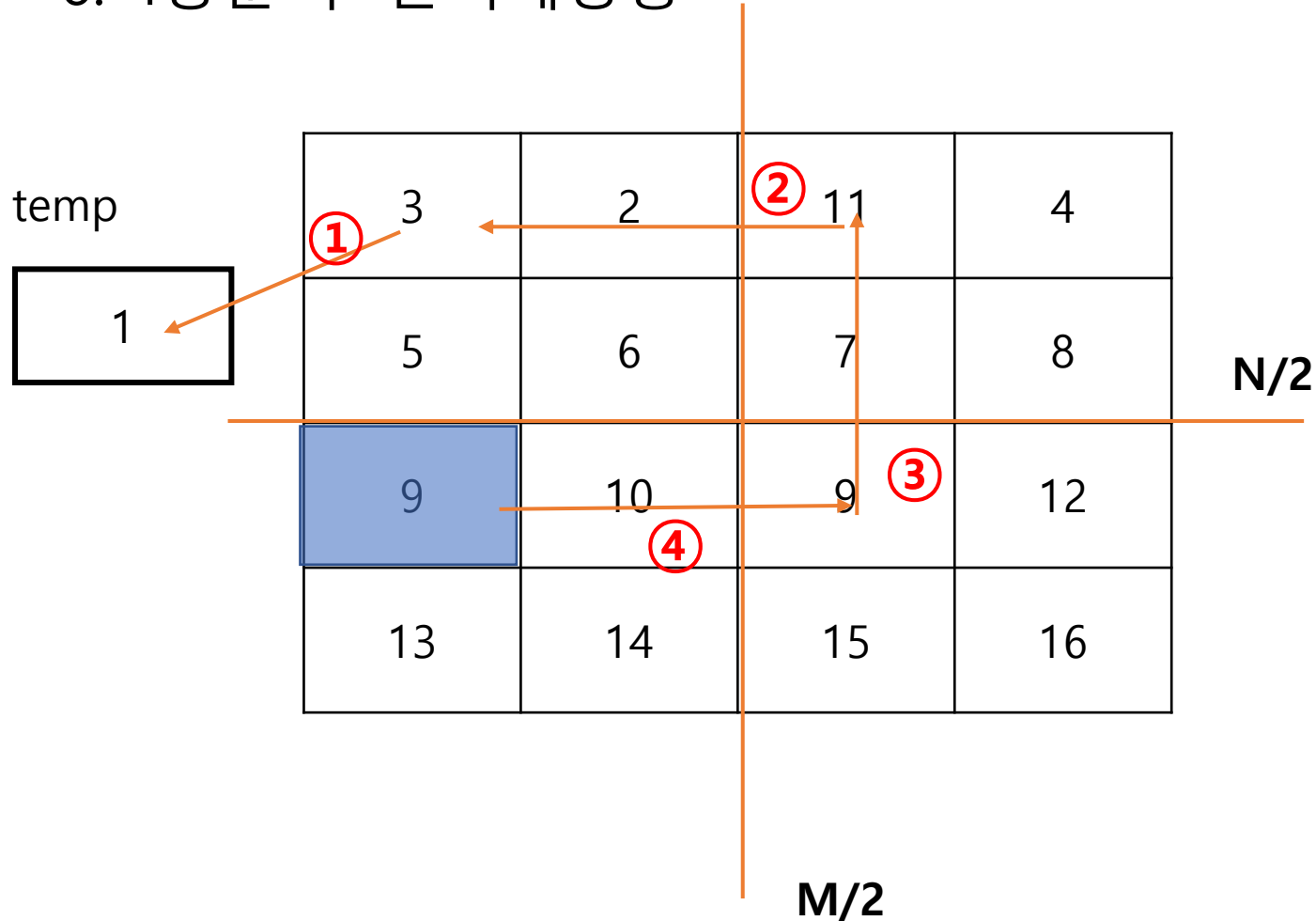
## 6. 4등분 후 반시계방향



N/2, M/2 범위 고정

# 16935-배열돌리기3

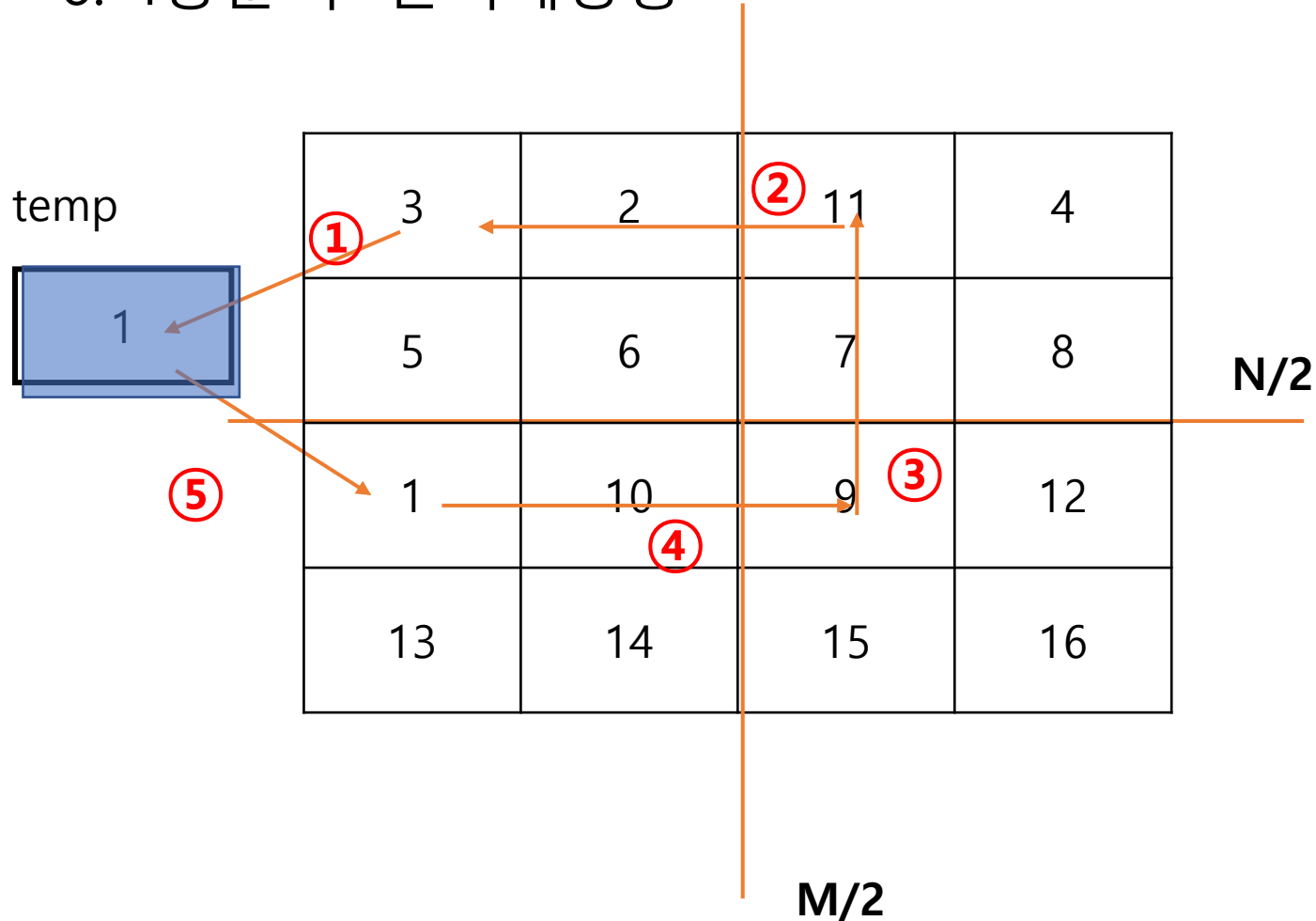
## 6. 4등분 후 반시계방향



N/2, M/2 범위 고정

# 16935-배열돌리기3

## 6. 4등분 후 반시계방향

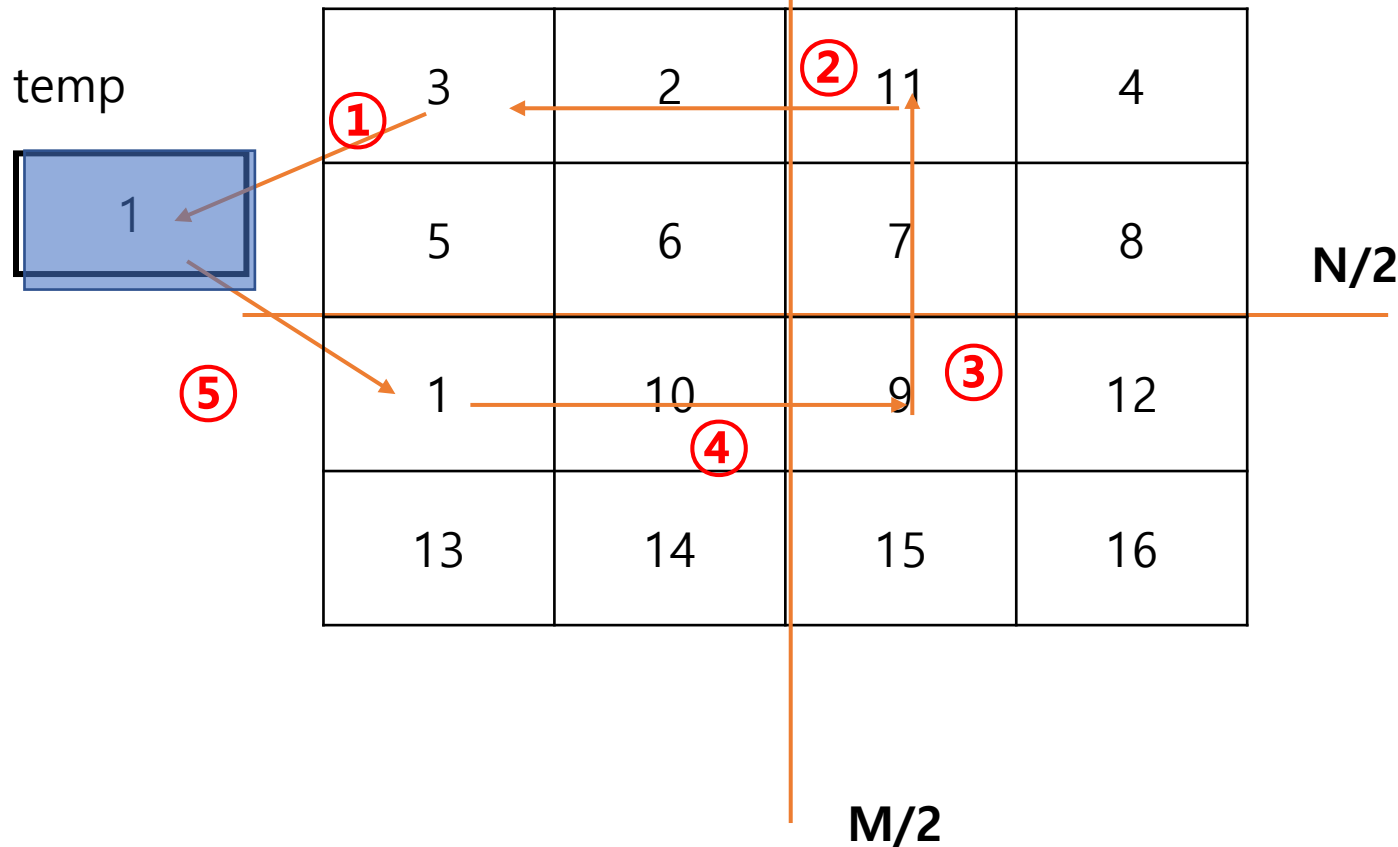


N/2, M/2 범위 고정



# 16935-배열돌리기3

## 6. 4등분 후 반시계방향



```
for (int i=0, h=N/2; i<h; i++){  
    for (int j=0, l=M/2; j<l; j++){  
        int tmp = map[i][j];  
        map[i][j] = map[i][j+l];  
        map[i][j+l] = map[i+h][j+l];  
        map[i+h][j+l] = map[i+h][j];  
        map[i+h][j] = tmp;  
    }  
}
```