

# Algorithm

---

- 배열 돌리기1

# 19626-배열돌리기1

## - 문제

크기가  $N \times M$ 인 배열이 있을 때, 배열을 돌려보려고 한다. 배열은 다음과 같이 반시계 방향으로 돌려야 한다.

```
A[1][1] ← A[1][2] ← A[1][3] ← A[1][4] ← A[1][5]
      ↓                               ↑
A[2][1]  A[2][2] ← A[2][3] ← A[2][4]  A[2][5]
      ↓       ↓           ↑       ↑
A[3][1]  A[3][2] → A[3][3] → A[3][4]  A[3][5]
      ↓                               ↑
A[4][1] → A[4][2] → A[4][3] → A[4][4] → A[4][5]
```

예를 들어, 아래와 같은 배열을 2번 회전시키면 다음과 같이 변하게 된다.

1 2 3 4	2 3 4 8	3 4 8 6
5 6 7 8	1 7 7 6	2 7 8 2
9 8 7 6	5 6 8 2	1 7 6 3
5 4 3 2	9 5 4 3	5 9 5 4
<시작>	<회전1>	<회전2>

### 제한

- $2 \leq N, M \leq 300$
- $1 \leq R \leq 1,000$
- $\min(N, M) \bmod 2 = 0$
- $1 \leq A_{ij} \leq 10^8$

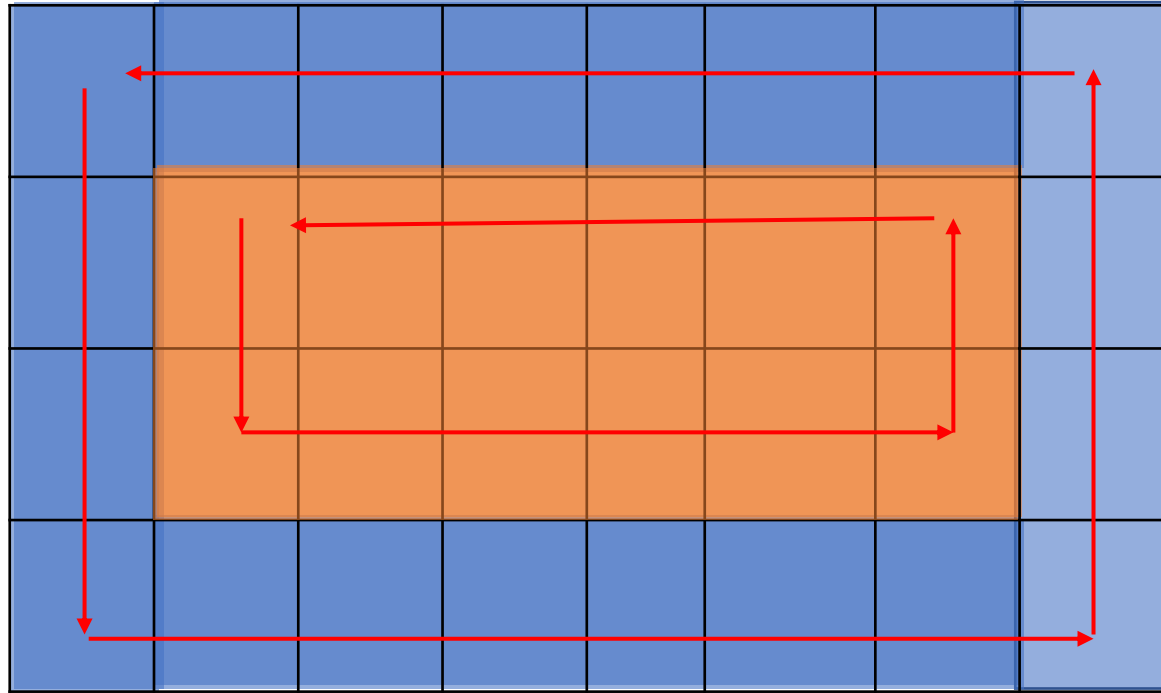
배열과 정수  $R$ 이 주어졌을 때, 배열을  $R$ 번 회전시킨 결과를 구해보자.

# 19626-배열돌리기1

## 제한

- $2 \leq N, M \leq 300$
- $1 \leq R \leq 1,000$
- $\min(N, M) \bmod 2 = 0$
- $1 \leq A_{ij} \leq 10^8$

$\min(N, M)/2$  : 돌려야 하는 사각형의 개수



# 19626-배열돌리기1

## 제한

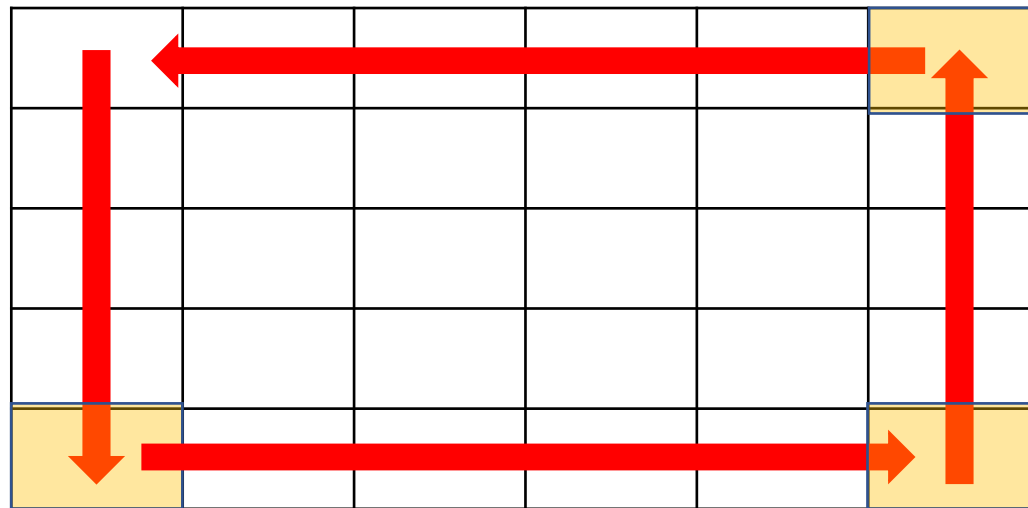
- $2 \leq N, M \leq 300$
- $1 \leq R \leq 1,000$
- $\min(N, M) \bmod 2 = 0$
- $1 \leq A_{ij} \leq 10^8$

$\min(N, M) \bmod 2$  : 회전하는 사각형 중 1줄만 회전하는 경우는 없음

--> 제한에 들어 왔다는 건, 입력도 저런 경우는 없음


# 19626-배열돌리기1

- 반시계 방향(하,우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기
- 진행 방향으로 값을 이동 시키는 경우
  - `map[0,0]` 값이 `map[1,0]`으로 이동 해야 함.
  - 이때, `map[1,0] = map[0,0]`으로 바로 값을 넣어주면, (1,0)위치의 값이 사라짐.
  - > 따라서, 이동하는 부분의 값을 임시로 담아 두어야함.



# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (0,0)

이동할 위치 : (1,0)

이전 값(temp)

1

이동할 위치 값(temp2)

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

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (1,0)

이동할 위치 : (1,0)

이전 값(temp)

1

이동할 위치 값(temp2)

7

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

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (1,0)

이동할 위치 : (1,0)

이전 값(temp)

1

이동할 위치 값(temp2)

7

1	2	3	4	5	6
1	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

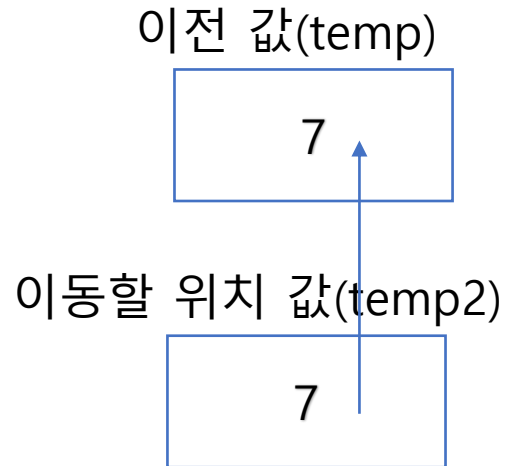


# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (1,0)

이동할 위치 : (1,0)



1	2	3	4	5	6
1	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (1,0)

이전 값(temp)

이동할 위치 : (2,0)

7

이동할 위치 값(temp2)

1	2	3	4	5	6
1	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (2,0)

이동할 위치 : (2,0)

이전 값(temp)

7

이동할 위치 값(temp2)

13

1	2	3	4	5	6
1	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (2,0)

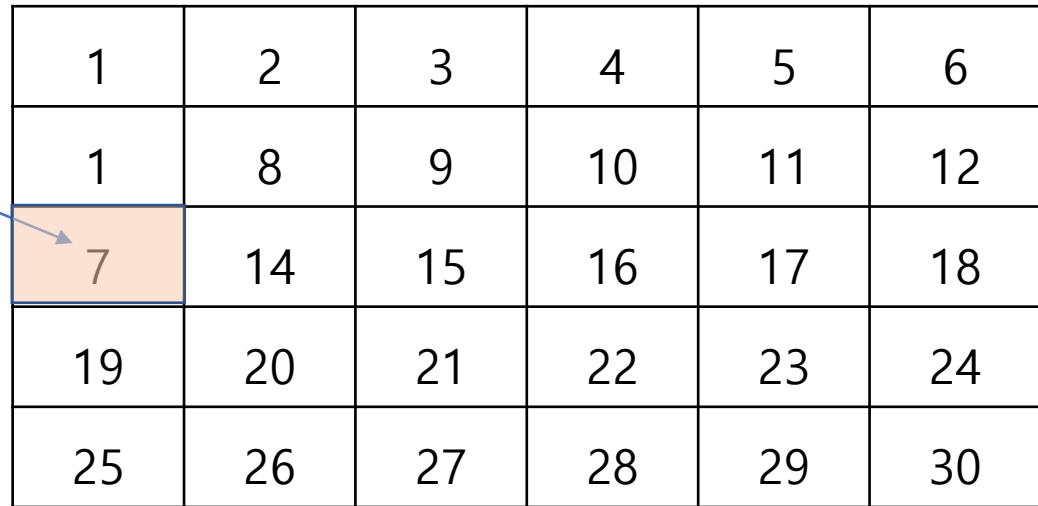
이동할 위치 : (2,0)

이전 값(temp)

7

이동할 위치 값(temp2)

13



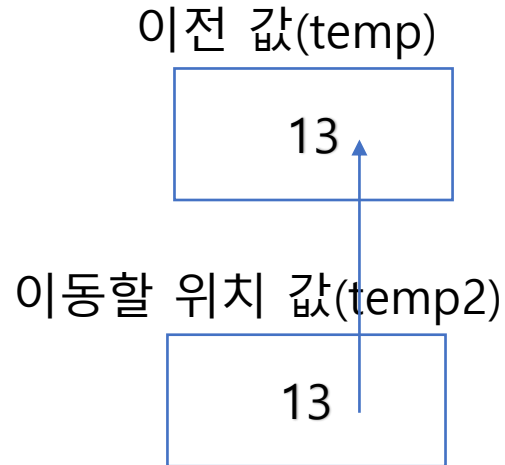
1	2	3	4	5	6
1	8	9	10	11	12
7	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (2,0)

이동할 위치 : (2,0)



1	2	3	4	5	6
1	8	9	10	11	12
7	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 1. 아래로 밀기(경계를 넘어가는 값은 방향 전환해서 넣어주기 위해 저장 해야함)

현재 위치 : (2,0)

이동할 위치 : (2,0)

이전 값(temp)

13

이동할 위치 값(temp2)

1	2	3	4	5	6
1	8	9	10	11	12
7	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

**\* 각 테두리마다 이 작업을 반복**

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (0,0)

가져올 값 위치 : (0,1)

시작 값(start)

1

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

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (0,0)

가져올 값 위치 : (0,1)

시작 값(start)

1

2	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



# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (0,1)

가져올 값 위치 : (0,1)

시작 값(start)

1

2	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

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (0,1)

가져올 값 위치 : (0,2)

시작 값(start)

1

2	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

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (0,1)

가져올 값 위치 : (0,2)

시작 값(start)



2	3	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

**\* 각 테두리마다 이 작업을 반복**

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (1,0)

가져올 값 위치 : (0,0)

시작 값(start)



2	3	4	5	6	12
7	8	9	10	11	18
7	14	15	16	17	24
13	20	21	22	23	30
19	25	26	27	28	29

**\*마지막에만 저장해둔 값 가져오기**

# 19626-배열돌리기1

- 반시계 방향(하, 우, 상, 좌)으로 돌리기 위한 코드 구현
  - 2. 오른쪽, 위, 왼쪽 순서로 모두 밀기 (도는 방향은 우, 하, 좌, 상)

현재 위치 : (1,0)

가져올 값 위치 : (0,0)

시작 값(start)

1



2	3	4	5	6	12
1	8	9	10	11	18
7	14	15	16	17	24
13	20	21	22	23	30
19	25	26	27	28	29

**\*마지막에만 저장해둔 값 가져오기**

**-> 다음 안쪽 테두리로!**