알튜비튜
정수론

오늘은 각종 수의 성질을 다루는 정수론에 대해 배웁니다.
특히, 최대공약수를 효율적으로 구하는 유클리드 호제법과 소수를 빠른 시간 내에 판별하는 에라토스테네스의 체에 대해 알아봅시다.

/<> 9421번 : 소수상근수 – Silver 1

문제

- 입력으로 주어진 양의 정수 n보다 작거나 같은 수들 중, 소수이면서 각 자리수의 제곱의 합을 반복적으로 계산했을 때 1이 되는 상근수인지 판단하는 문제

제한 사항

- 입력으로 주어지는 n : 10 ≤ n ≤ 1000000

예제 입력1

```
20
```

예제 출력1

```
7
13
19
```

소수상근수란?

소수상근수

- 소수이면서 상근수인 수
- 1부터 n 사이의 모든 정수들에 대해 소수이면서 상근수인지 판단
  -> 반복문 이용!

- n개의 숫자들에 대해 각각 에라토스테네스의 체를 적용하는 것은 비효율적
  -> 1부터 n 사이의 소수부터 먼저 찾고
  -> 여기서 찾은 소수들에 대해서만 상근수인지 판단

소수 찾는 방법을 복습해볼까요?

#알튜비튜

에라토스테네스의 체

- 각 수가 소수인지 판단한 여부를 저장하는 배열 사용
- 2부터 시작해서 해당 숫자의 배수에 해당하는 숫자들을 지워나감 (~ $\sqrt{n}$)
  -> 약수가 존재하면 소수가 아니므로
  -> 해당 숫자는 소수

- 여기서 구한 숫자들에 대해서만 상근수 여부 판단해주기

상근수 찾기

#알튜비튜

상근수

- 각 자리수의 제곱의 합을 반복적으로 구했을 때 1이 될 수 있는 수

상근수 찾기

- 700 → $7^2 + 0^2 + 0^2 = 49$
- 49 → $4^2 + 9^2 = 97$
- 97 → $9^2 + 7^2 = 130$
- 130 → $1^3 + 3^2 + 0^2 = 10$
- 10 → $1^2 + 0^2 = 1$

상근수 찾기

- 700 → $7^2 + 0^2 + 0^2 = 49$
- 49 → $4^2 + 9^2 = 97$
- 97 → $9^2 + 7^2 = 130$
- 130 → $1^3 + 3^2 + 0^2 = 10$
- 10 → $1^2 + 0^2 = 1$

상근수 찾기

- 700 → $7^2 + 0^2 + 0^2 = 49$
- 49 → $4^2 + 9^2 = 97$
- 97 → $9^2 + 7^2 = 130$
- 130 → $1^3 + 3^2 + 0^2 = 10$
- 10 → $1^2 + 0^2 = 1$

∴ 700은 상근수

상근수 찾기

- $2 \rightarrow 2^2 = 4$
- $4 \rightarrow 4^2 = 16$
- $16 \rightarrow 1^2 + 6^2 = 37$
- $37 \rightarrow 3^2 + 7^2 = 58$
- $58 \rightarrow 5^2 + 8^2 = 89$
- $89 \rightarrow 8^2 + 9^2 = 145$
- $145 \rightarrow 1^2 + 4^2 + 5^2 = 42$
- $42 \rightarrow 4^2 + 2^2 = 20$
- $20 \rightarrow 2^2 + 0^2 = 4$
- $4 \rightarrow 4^2 = 16$
- $\cdots$

상근수 찾기

- $2 \rightarrow 2^2 = 4$
- $4 \rightarrow 4^2 = 16$
- $16 \rightarrow 1^2 + 6^2 = 37$
- $37 \rightarrow 3^2 + 7^2 = 58$
- $58 \rightarrow 5^2 + 8^2 = 89$
- $89 \rightarrow 8^2 + 9^2 = 145$
- $145 \rightarrow 1^2 + 4^2 + 5^2 = 42$
- $42 \rightarrow 4^2 + 2^2 = 20$
- $20 \rightarrow 2^2 + 0^2 = 4$
- $4 \rightarrow 4^2 = 16$
- $\cdots$

상근수 찾기

- $2 \rightarrow 2^2 = 4$
- $4 \rightarrow 4^2 = 16$
- $16 \rightarrow 1^2 + 6^2 = 37$
- $37 \rightarrow 3^2 + 7^2 = 58$
- $58 \rightarrow 5^2 + 8^2 = 89$
- $89 \rightarrow 8^2 + 9^2 = 145$
- $145 \rightarrow 1^2 + 4^2 + 5^2 = 42$
- $42 \rightarrow 4^2 + 2^2 = 20$
- $20 \rightarrow 2^2 + 0^2 = 4$
- $4 \rightarrow 4^2 = 16$
- $\cdots$

$\therefore$ 2는 상근수가 아님!

상근수 찾기

- 상근수인지 아닌지 어떻게 판단할까?
- 상근수가 아닌 경우 무한대로 계산이 반복됨
  -> 앞의 연산에서 나왔던 수가 반복적으로 나타남

- 이미 한 번 나왔던 결과가 다시 나온다면 해당 수는 상근수가 아님!
  -> 상근수를 판단하는 연산 과정에서 나왔던 수를 set에 저장
  -> 이번 연산 결과가 이미 set에 있다면 상근수가 아니라고 판단하기

도전 문제 2

2981번 : 검문 - Gold 4

문제

- N개의 숫자들을 각각 나눴을 때의 나머지들이 모두 같게 되는 M을 찾는 문제

제한 사항

- 종이에 적은 수의 개수 N : 2 ≤ N ≤ 100
- 종이에 적은 수는 모두 1보다 크거나 같고, 1,000,000,000보다 작거나 같은 자연수

예제

예제 입력1

```
3
6
34
38
```

예제 출력1

```
2 4
```

예제 입력1

```
5
5
17
23
14
83
```

예제 출력1

```
3
```

나머지가 모두 같게 되는 M…?

- N개의 수를 1부터 1,000,000,000 각각 나누면서 판단해줄까요…?
  -> O(N) = O(100 * 1,000,000,000)
  -> 너무 비효율적

- M이 가지는 특징을 살펴봅시다!

나머지가 모두 같게 되는 M…?

#알튜비튜

- A, B, C를 M으로 나눴을 때의 나머지가 모두 R이라면?
  -> A = M * a + R
  B = M * b + R
  C = M * c + R (a, b, c는 M으로 나눴을 때의 몫)

- B – A = (M * b + R) – (M * a + R) = M * (b – a)
- A – C = (M * a + R) – (M * c + R) = M * (a – c)

나머지가 모두 같게 되는 M…?

#알튜비튜

- A, B, C를 M으로 나눴을 때의 나머지가 모두 R이라면?
  -> A = M * a + R
  B = M * b + R
  C = M * c + R (a, b, c는 M으로 나눴을 때의 몫)

- B – A = (M * b + R) – (M * a + R) = M * (b – a)
- A – C = (M * a + R) – (M * c + R) = M * (a – c)
  -> 각 수의 차가 모두 M에 대해 나누어 떨어짐!

나머지가 모두 같게 되는 M…?

- A, B, C를 M으로 나눴을 때의 나머지가 모두 R이라면?
  -> A = M * a + R
  B = M * b + R
  C = M * c + R (a, b, c는 M으로 나눴을 때의 몫)

- B – A = (M * b + R) – (M * a + R) = M * (b – a)
- A – C = (M * a + R) – (M * c + R) = M * (a – c)
  -> 각 수의 차가 모두 M에 대해 나누어 떨어짐!

  -> M = 이웃한 수들(B – A, C – B) 간의 차의 모든 공약수

모든 M은 어떻게 찾을까요?

- M = 이웃한 수들(B – A, C – B) 간의 차의 모든 공약수

- 이웃한 수들 간의 차에 대해 먼저 최대공약수(GCD)를 구해주고
- 이 최대공약수(GCD)의 모든 약수를 구해주면
- 모든 M을 구할 수 있음!

최대공약수 구하는 법을 복습해볼까요?

#알튜비튜

유클리드 호제법

- $A = a \cdot G$
- $B = b \cdot G$ (a와 b는 서로소)
- $A = q \cdot B + r$ (q = A/B 의 몫, r = A%B)
- r (A%B) = $a \cdot G - q \cdot b \cdot G = (a - q \cdot b) \cdot G$

-> $(a - q \cdot b)$ 와 b 또한 서로소 이므로 A%B 와 B 의 최대공약수도 G

- GCD(A, B) = GCD(A-B, B) = GCD(A-2B, B) = ⋯ = GCD(A%B, B)

2840번 : 행운의 바퀴 – Silver 4

문제

- 바퀴의 회전 수와, 회전이 끝난 후 가리키는 글자가 주어졌을 때, 행운의 바퀴를 구하자.
- 빈 칸에 어떤 글자가 들어갈 지 알 수 없으면 ?로 출력
- 행운의 바퀴를 만들 수 없으면 ! 출력하고 끝내기

제한 사항

- 바퀴의 칸 수 N: 2 <= N <= 25
- 바퀴를 돌리는 횟수 K: 1 <= K <= 100
- 바퀴에 같은 글자는 두 번 이상 등장 X
- 바퀴는 시계 방향으로 돌아감
- +) 원판의 한 칸에 글자 2개 이상 못 들어감

예제

예제 입력1

```
3 3
1 A
2 B
3 C
```

예제 입력2

```
5 6
1 A
2 B
5 B
1 C
2 A
2 B
```

예제 입력3

```
8 8
4 V
3 I
7 T
7 A
6 R
5 N
1 O
9 H
```

예제 출력1

```
!
```

예제 출력2

```
B?A?C
```

예제 출력3

```
HONITAVR
```

잠깐

- 알파벳이 여러번 쓰였는지 중복 체크

→ 알파벳 사용 여부 관리 배열: is_available

- 원판의 한 칸에 문자가 2개 이상 들어갈 수 없음

→ 원판의 해당 위치에 알파벳 없으면 OK
→ 원판의 해당 위치에 알파벳이 있는데, 그게 자기 자신과 동일하면 OK
→ 원판의 해당 위치에 알파벳이 있는데, 그게 다른 글자이면 원판 만들기 불가능

잠깐

- 바퀴를 배열로 보고 시계방향으로 바퀴를 회전시키면, 화살표가 가리키고 있는 배열의 원소는 인덱스가 감소하는 것처럼 보임

→ 이 과정에서, 시계 방향을 배열의 -/+ 방향 중 하나로 선택할 수 있음

- 1) 시계 방향을 배열의 - 방향으로 두는 경우 (문제 조건과 같은 방향)

→ 바퀴를 돌릴 때마다 인덱스가 감소함

→ 바퀴 회전 횟수에 따라 인덱스가 음수가 되는 경우 발생

- 2) 시계 방향을 배열의 + 방향으로 두는 경우 (문제 조건과 반대 방향)

→ 바퀴를 돌릴 때마다 인덱스가 증가함

→ 인덱스가 음수가 되는 상황이 발생하지 않도록, + 방향 선택

→ 마지막에 바퀴에 적어놓은 알파벳을 출력할 때는 돌릴 때와 반대 방향으로 출력

예제1

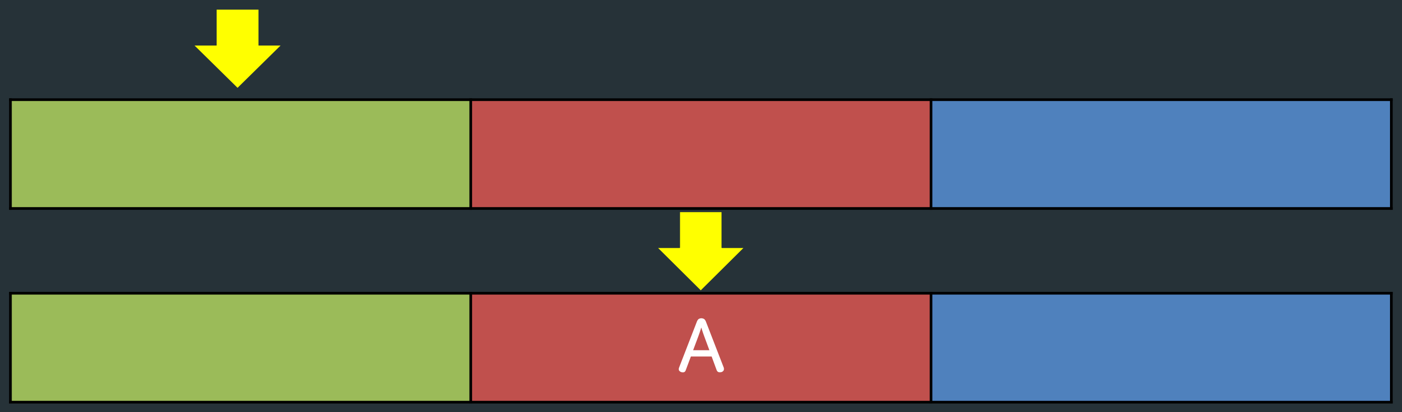
예제 입력

```
3 3
1 A
2 B
3 C
```

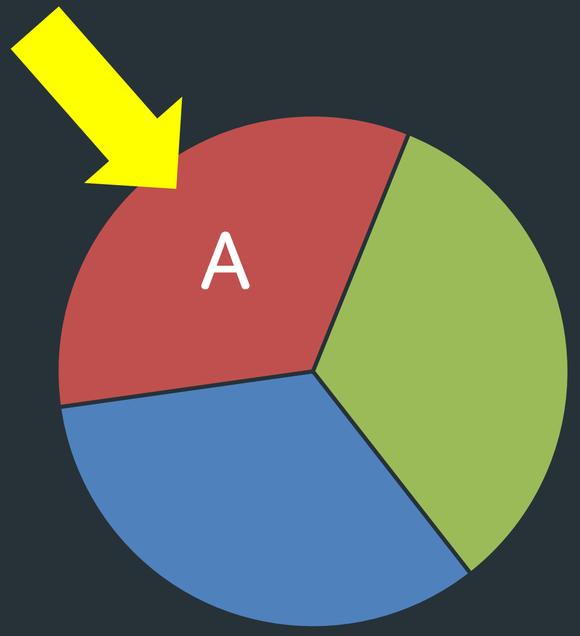
예제 출력

```
!
```

START



START



1

예제1

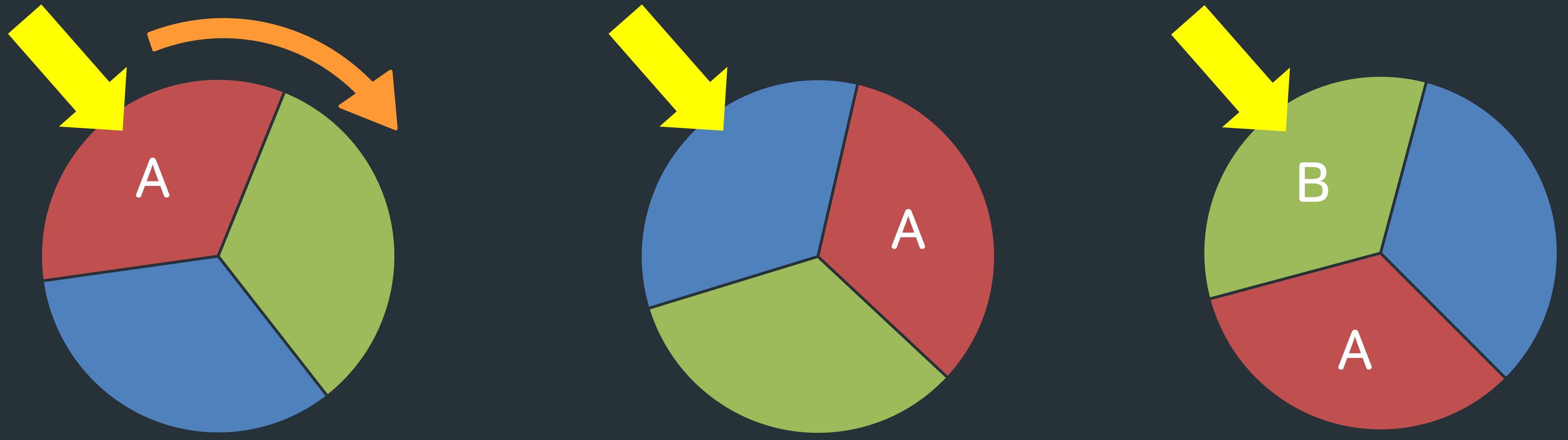
예제 입력

```
3 3
1 A
2 B
3 C
```

예제 출력

```
!
```

START



START 1 2

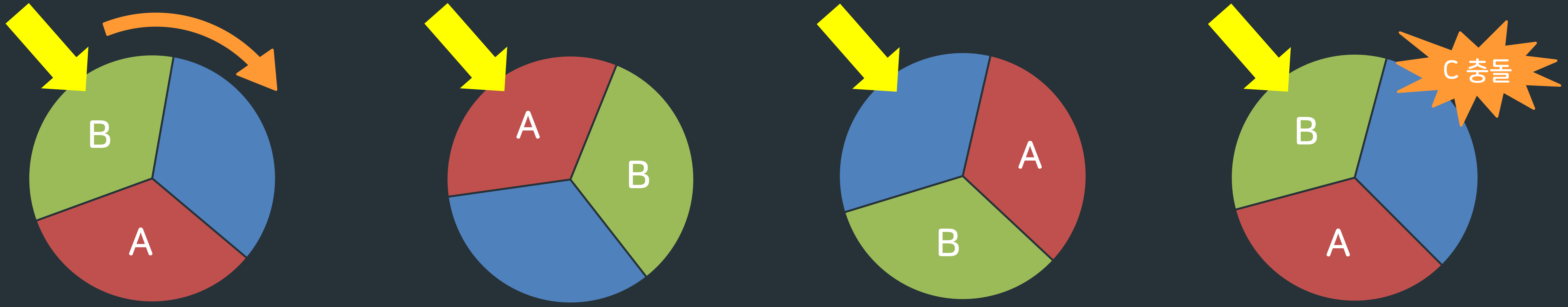
예제1

예제 입력

```
3 3
1 A
2 B
3 C
```

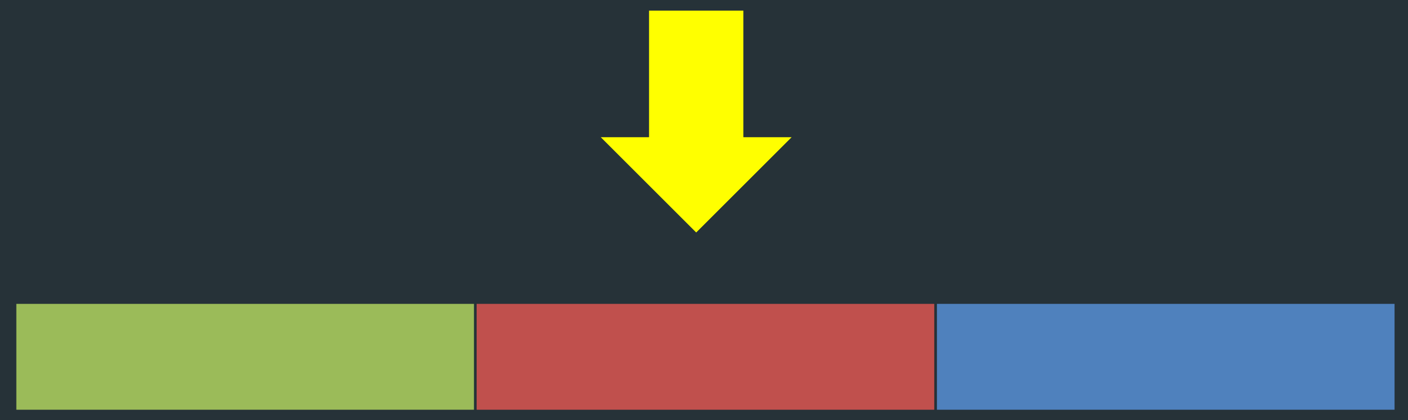
예제 출력

```
!
```



START 1 2 3

정리하자면...



화살표가 바퀴가 돌아가는 횟수만큼 우측으로 이동
나머지 연산으로 인덱스를 범위 안으로 만들기

index = (index + s) % n

마무리

추가로 풀어보면 좋은 문제!

- 14490번 : 백대열 – Silver 5
- 9613번 : GCD 합 – Silver 4
- 2168번 : 타일 위의 대각선 – Silver 1
- 20302번 : 민트 초코 – Gold 4