



# 피자 나눠 먹기(하)

|         |                             |
|---------|-----------------------------|
| 종료일     | @2023년 5월 29일               |
| 태그      | arithmetic java programmers |
| 상태      | Done                        |
| 작업 시간   | 10                          |
| # Level | 0                           |

## 제목 입력

프로그래머스 알고리즘 문제 URL

<https://school.programmers.co.kr/learn/courses/30/lessons/120814>

## 문제 설명

### ▼ 문제

머쓱이네 피자가게는 피자를 일곱 조각으로 잘라 줍니다. 피자를 나눠먹을 사람의 수  $n$ 이 주어질 때, 모든 사람이 피자를 한 조각 이상 먹기 위해 필요한 피자의 수를 return 하는 solution 함수를 완성해보세요.

## 제한사항

### ▼ 제한사항

- $1 \leq n \leq 100$

## 입출력 예

### ▼ 입출력 예

| n  | result |
|----|--------|
| 7  | 1      |
| 1  | 1      |
| 15 | 3      |

### ▼ 입출력 예 설명

입출력 예 #1

- 7명이 최소 한 조각씩 먹기 위해서 최소 1판이 필요합니다.

입출력 예 #2

- 1명은 최소 한 조각을 먹기 위해 1판이 필요합니다.

입출력 예 #3

- 15명이 최소 한 조각씩 먹기 위해서 최소 3판이 필요합니다.

## 풀이

### ▼ 풀이과정

산수 문제 피자는 7조각으로 나눠주고 인당 하나씩 먹기 위해서 7을 기준으로 몫과 나머지 여부를 판단하여 필요한 총 피자 개수 파악

몫 : `n / 7` , 나머지 : `n % 7`

### ▼ 테스트 전체 코드

내용

```
package level_0.sharingOne;

public class Main {
    public static void main(String[] args) throws Exception {
        Solution solution = new Solution();
        System.out.println("결과 값" + solution.solution(8));
    }

    public static class Solution{
        // TODO 피자 나눠 먹기
        // INFO : 피자는 일곱 조각으로 잘라준다. 나눠먹는 사람의 수 n 일때 모든 사람이 피자를 한조각 이상 먹기 위해 필요한 피자의 수 return
        public int solution(int n) {
            boolean val = (0 < n && n <= 100);
            int answer = 0;
            System.out.printf("몫 : %d\t나머지 : %d\n", n/7, n % 7);
            //answer = val ? n/7 + (n % 7 > 0 ? 1 : 0) : 0;
            // answer = val ? n % 7 == 0 ? n / 7 : n / 7 + 1 : 0;
            answer = val ? (n + 6)/7 : 0;
            return answer;
        }
    }
}
```

### ▼ 다른 사람 풀이

```
class Solution {
    public int solution(int n) {
        return (n + 6) / 7;
    }
}
```

```
class Solution {
    public int solution(int n) {
        int answer = 0;

        int diviNum = n / 7;
        int restNum = n % 7;

        if (restNum > 0) {
            diviNum += 1;
        }

        answer = diviNum;
    }
}
```

```
        return answer;
    }
}
```

```
class Solution {
    public int solution(int n) {
        int answer = n/7;
        if(!(n%7==0)) answer++;
        return answer;
    }
}
```

---

## 느낀점

문제 해석은 같으나 풀이 코드 방식들이 여러가지 있는걸 알게됨