

# 알고리즘 과제 3

## The gift Solution

## 01. 문제 설명

필리피우스는 그들 중 한 사람에게 선물을 사기 위해 서로 다른 예산을 모아야 합니다.

하지만 각자의 예산이 다르기 때문에 **가장 공정한 방법**으로 각자가 얼마를 부담해야 할지 고민하고 있습니다.

이들은 며칠 동안 논의했지만 **모두가 만족할 만한 해결책**을 찾지 못했습니다.

이에 **닥터**가 프로그램을 만들어 해결책을 제시하기로 했습니다.

## 02. 조건

여러 사람이 각자의 **최대 예산 내**에서 **공정하게 비용을 분담해**  
**선물의 가격**을 충당해야 합니다.

**누구도 자신의 최대 예산을 초과해서 지불할 수 없습니다.**

**가장 많이 기여한 금액이 최소화**되도록 나눕니다.

**모든 기여금**은 **정수**여야 합니다.

## 03. 출력, 입력

선물 가격을 충당할 수 있으면 **기여금을 오름차순**으로 출력합니다.

**충당이 불가능하면 IMPOSSIBLE을 출력합니다.**

- ✓ 사람의 수  $N$  ( $1 \leq N \leq 2000$ )
- ✓ 선물의 가격  $C$  ( $0 \leq C \leq 1,000,000,000$ )
- ✓ 각 참가자의 최대 예산  $B$  ( $0 \leq B \leq 1,000,000,000$ )

## 04. 예제

✓ Input 1

N : 3, C : 100, B : 20, 20, 40 ▶ 예산에 못 미침, IMPOSSIBLE

✓ Input 2

N : 3, C : 100, B : 40, 40, 40 ▶ 33 33 34

✓ Input 3

N : 3, C : 100, B : 100, 1, 60 ▶ 1 49 50

## 05. 풀이과정

✓ 버블정렬로도 풀이가 가능하나, 본인은 퀵정렬을 사용하였음.

```
int main() {
    int N, price;
    scanf("%d %d", &N, &price); // 참여자 수와, 선물 가격을 읽음

    int budgets[N]; // 각 참여자의 예산을 저장할 배열
    int totalBudget = 0; // 참여자 전체 예산의 총합

    // 각 참여자의 예산을 입력받아 배열에 저장하고, 총 예산 계산
    for (int i = 0; i < N; i++) {
        scanf("%d\n", &budgets[i]);
        totalBudget += budgets[i];
    }
}
```

## 05. 풀이과정

✓ 버블정렬로도 풀이가 가능하나, 본인은 퀵정렬을 사용하였음.

```
if (totalBudget < price) // 총 예산이 선물 가격보다 적으면 불가능 (IMPOSSIBLE)
    printf("IMPOSSIBLE\n");
else {
    /* 예산 배열을 정렬한 후, 작은 예산부터 시작하여
       각 참여자가 (남은 가격) / (남은 참여자 수)를 기여하거나,
       자신의 예산이 부족하면 가능한 최대 금액을 기여함 */
    qsort(budgets, N, sizeof(int), cmp); // 예산 배열을 오름차순 정렬
    int contribution; // 각 참여자의 기여금

    for (int i = 0; i < N; i++) {
        /* 기여금은 남은 가격을 남은 참여자 수로 나눈 값이거나,
           해당 참여자의 최대 예산 중 더 작은 값을 선택 */
        contribution = (budgets[i] >= price / (N - i)) ? price / (N - i) : budgets[i];
        printf("%d\n", contribution); // 기여금 출력
        price -= contribution; // 남은 선물 가격에서 기여금을 차감
    }
}
```

## 06. 문제 풀 때 힘들었던 점

가장 힘들었던 건, 기여금을 나눌 때가 제일 힘들었음.

예를 들어, 산술 연술자를 어떻게 사용해야 했을까?

```
contribution = (budgets[i] >= price / (N - i) ) ? price  
/ (N - i) : budgets[i];
```