

## 당직표 짜기

이 문제는 2조의 문제입니다.

제출 내용:

- 주제 (프로그램 구현 가능한 주제)

당직표 짜기

- 누구에게 도움이 되는가? (Contribution)

군대나 회사에서 직급에 관계 없이 평등하게 당직을 짜려는 사람들에게

- Decrease and Conquer 방식 중 어느 것에 해당하는가?

Josephus Problem Decrease by constant factor

🔴 왜 그 방식에 해당하는가?

constant factor의 배수 만큼 순회하면서 한 명씩 뽑아내기 때문에 중복되지 않고 한바퀴를 돌 수 있음

- 알고리즘 정의 (IOP)

INPUT : 총 인원 수, 사람 명단, constant factor

OUTPUT : 평일 당직이 1번씩 배정된 당직 인원 명단

평일 당직과 주말 당직을 구할 때는 각각 다른 배열과 constant factor을 사용해서 알고리즘을 진행한다.

파이썬을 이용하여 구현하였습니다. 다음의 명단을 기반으로 평일과 주말의 당직표를 짜는 프로그램을 만들었습니다. 당직표의 시작 요일을 월요일이라 가정하였습니다.

```
names = [
    '민석', '창희', '태완', '연준', '용주', '보민',
    '소미', '서준', '예준', '도윤', '시우', '주원',
    '하준', '지호', '지후', '서연', '서윤', '지우',
    '서현', '하윤', '윤서', '지유', '지민', '채원',
    '지윤', '은서', '수아', '다은', '예은', '지아'
]
```

```

PS C:\Users\alexh\Desktop\Algorithm\week8> & C:/Users/alexh/AppData/Local/Microsoft/Windows
<<< 평일 당직표 >>>

월요일: 태완 지우 연준 은서 다른 지민
화요일: 보민 윤서 서준 민석 용주 시우
수요일: 예준 채원 하준 소미 서연 창희
목요일: 주원 수아 서윤 지호 지윤 서현
금요일: 지후 지아 지유 하윤 도윤 예은

<<< 주말 당직표 >>>

토요일: 태완 예준 지후 윤서 수아 연준 하준 지유 민석 지호 다른 서연 도윤 시우 서현
일요일: 보민 주원 지우 채원 지아 서준 서윤 은서 소미 하윤 용주 지윤 지민 창희 예은

```

코드는 다음과 같습니다.

```

names = [
    '민석', '창희', '태완', '연준', '용주', '보민',
    '소미', '서준', '예준', '도윤', '시우', '주원',
    '하준', '지호', '지후', '서연', '서윤', '지우',
    '서현', '하윤', '윤서', '지유', '지민', '채원',
    '지윤', '은서', '수아', '다른', '예은', '지아'
]
count = len(names) # 30
weekdays_ids = [i for i in range(count)]
weekends_ids = [i for i in range(count)]
constant = 3 # random.randint(1, 10)

members_order = []

# joshephus problem 을 이용하여 풀려하였으나, 그렇게 하면 과정이
# 출력되지 못해 다른 방식으로 구현합니다.

# 뽑히는 순서를 members_order 에 차례대로 담아주기로 하였습니다.
def modified_josephus(arr, k, cursor):
    newCount = len(arr)
    newCursor = (cursor+k-1) % newCount
    if len(arr) == 1:
        return members_order.append(arr.pop(newCursor))
    members_order.append(arr.pop(newCursor))
    return modified_josephus(arr, k, newCursor)

modified_josephus(weekdays_ids, constant, 0)

# 평일 당번과 주말 당번을 따로 따로 담아주었습니다.

```

```

weekdays_order = [[] for i in range(5)]
weekends_order = [[] for i in range(2)]

for i in range(count):
    if i%5 == 0:
        weekdays_order[0].append(members_order[i])
    elif i%5 == 1:
        weekdays_order[1].append(members_order[i])
    elif i%5 == 2:
        weekdays_order[2].append(members_order[i])
    elif i%5 == 3:
        weekdays_order[3].append(members_order[i])
    elif i%5 == 4:
        weekdays_order[4].append(members_order[i])

for i in range(count):
    if i%2 == 0:
        weekends_order[0].append(members_order[i])
    elif i%2 == 1:
        weekends_order[1].append(members_order[i])

# 평일 당직표 출력해주기
print("<<< 평일 당직표 >>>\n")
for i in range(5):
    if i%5 == 0:
        print("월요일: ", end='')
        for j in range(len(weekdays_order[0])):
            print(names[weekdays_order[0][j]], end=' ')
        print("\n")
    elif i%5 == 1:
        print("화요일: ", end='')
        for j in range(len(weekdays_order[1])):
            print(names[weekdays_order[1][j]], end=' ')
        print("\n")
    elif i%5 == 2:
        print("수요일: ", end='')
        for j in range(len(weekdays_order[2])):
            print(names[weekdays_order[2][j]], end=' ')
        print("\n")
    elif i%5 == 3:
        print("목요일: ", end='')
        for j in range(len(weekdays_order[3])):
            print(names[weekdays_order[3][j]], end=' ')
        print("\n")
    elif i%5 == 4:
        print("금요일: ", end='')
        for j in range(len(weekdays_order[4])):
            print(names[weekdays_order[4][j]], end=' ')

```

```

        print("\n")

# 주말 당직표 출력해주기
print("\n<<< 주말 당직표 >>>\n")
for i in range(2):
    if i%2 == 0:
        print("토요일: ", end='')
        for j in range(len(weekends_order[0])):
            print(names[weekends_order[0][j]], end=' ')
        print("\n")
    else:
        print("일요일: ", end='')
        for j in range(len(weekends_order[1])):
            print(names[weekends_order[1][j]], end=' ')

```

Josephus를 살짝 변형하여 마지막으로 살아남는 사람만을 구하는 것이 아닌, 마지막으로 살아남는 사람이 나올 때까지 제거되는 사람도 따로 리스트(members\_order)에 담는 방식으로 구현하였습니다.