

## HOMEWORK 1 - Population counters

### Zadání :

- Uděláme počítač obyvatel pro obce A, B, C a D
- Obce a počet obyvatel máme uložené ve slovníku
  - towns = { 'A': 150, 'B': 279, 'C': 869, 'D': 974 }
- Vytvořte funkci born, která veme jako argument seznam obcí, kde se narodilo dítě
- Funkce died má obdobně zemřelé obyvatele odečítat
- Funkce berou argumenty listu nebo budou fungovat pomocí \*args (výběr je na vás)
  - příklad: born('A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D') nebo born(['A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D'])

### Řešení :

```
towns = { 'A': 150, 'B': 279, 'C': 869, 'D': 974 }
```

```
# first alternative
def born (*towns_borned):
    for t in towns_borned:
        towns[t] += 1

def died (*towns_died):
    for t in towns_died:
        towns[t] -= 1

born('A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D')
died('A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D')
```

```
# second alternative
def born (towns_borned):
    for t in towns_borned:
        towns[t] += 1

def died (towns_died):
    for t in towns_died:
        towns[t] -= 1

born(['A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D'])
died(['A', 'B', 'A', 'A', 'C', 'D', 'C', 'D', 'C', 'D'])
```

## HOMEWORK 2 - Calendar Application

## Zadání :

- Máme týdenní kalendář uložený ve slovníku, kde klíče jsou jednotlivé dny a hodnoty jsou listy o délce 24, pro každou hodnotu jeden element
- Definice takového kalendáře vypadá (False hodnota značí prázdné místo):
  - ```
week_calendar = {  
    'mon': [False] * 24,  
    'tue': [False] * 24,  
    'wen': [False] * 24,  
    'thu': [False] * 24,  
    'fri': [False] * 24,  
}
```
- Na obsluhu napište následující funkce:
  - `check_free(day, hour)` -> vrátí True pokud je v dané době volno (volno = False)
  - `add_event(event, day, hour_from, hour_to)` -> přidá do daného rozmezí novou událost, pokud tam žádná není
  - `clear_hour(day, hour)` -> vymaže událost na daném místě

## Řešení :

```
week_calendar = {  
    'mon': [False] * 24,  
    'tue': [False] * 24,  
    'wen': [False] * 24,  
    'thu': [False] * 24,  
    'fri': [False] * 24,  
}  
  
def check_free(day, hour):  
    if (week_calendar[day][hour] == False):  
        return True  
    else:  
        return False  
  
def clear_hour(day, hour):  
    week_calendar[day][hour] = False  
  
def add_event(event, day, hour_from, hour_to):  
    # checking if hours are free  
    free = True  
    for hour in range(hour_from, hour_to):  
        if (check_free(day, hour) == False): # using function check_free  
            free = False  
  
    # inserting event  
    if free == True: # also "if free:" works  
        for hour in range(hour_from, hour_to):  
            week_calendar[day][hour] = event  
  
# testing  
add_event('event1', 'mon', 10, 16)  
add_event('event2', 'mon', 14, 18)
```

```
clear_hour('mon', 13)  
print(week_calendar)
```