

נתונה השאלה הבאה:

כיתבו פונקציה גנרטור בשם `mult_sequence` שמקבלת זוג ארגומנטים: איטרטור `values_it` ואיטרטור `counts_it`, ומחזירה גנרטור שמוגדר באופן הבא:
נניח ש: `values_it` מייצר את האיברים a_1, a_2, a_3, \dots
וכן, `counts_it` מייצר את האיברים n_1, n_2, n_3, \dots
אז הגנרטור שמוחזר ע"י `mult_sequence` ייצר את הרצף הבא:

$$\underbrace{a_1, a_1, \dots, a_1}_{n_1 \text{ times}}, \underbrace{a_2, a_2, \dots, a_2}_{n_2 \text{ times}}, \underbrace{a_3, a_3, \dots, a_3}_{n_3 \text{ times}}, \dots$$

לדוגמא:

```
>>> values = ["a","b","c"]
>>> counts = [2,0,1]
>>> list(mult_sequence(iter(values),iter(counts)))
["a", "a", "c"]
>>> gen = mult_sequence(iter(values),iter(counts))
>>> next(gen)
"a"
```

הערות:

1. האיטרטורים בקלט אינם בהכרח סופיים.
2. אם אחד האיטרטורים הגיע לסיומו על הגנרטור לזרוק `StopIteration exception`.

פתרון השאלה:

מאחר שהפונקציה אמורה להחזיר גנרטור עלינו לכתוב פונקציית גנרטור (כלומר על הפונקציה להכיל פקודות `yield`).

בשלב ה i נקלוט את האיבר ה i שמיוצר ע"י `values_it` (קריאה ל `next()`) לתוך משתנה `next_val`, וכמו כן נקלוט את האיבר ה i שמיוצר ע"י `counts_it` לתוך משתנה `next_count`.

כעת נייצר את הערך `next_val` מספר `next_count` פעמים באמצעות לולאה שמבצעת `next_count` איטרציות.

מכיוון שייתכן שהאיטרטורים `values_it`, `counts_it` אינם סופיים יהיה עלינו לבצע את התהליך המתואר בתוך לולאה אינסופית:

```
while (true):
```

```
.....
```

אם בשלב כלשהו אחד האיטרטורים הגיע לסיומו תיווצר StopIteration Exception , כפי שצויין בהערה (2).

```
def mult_sequence (values_it, counts_it):  
    while(True):  
        next_val = next(values_it)  
        next_count = next(counts_it)  
        for i in range(next_count):  
            yield next_val
```