

תרגיל 5.4

התבקשתם לסייע לרשות האוכלוסין לכתוב תוכנית שמחוללת מספרים לתעודות זהות - תוכנית שמקצה מספר תעודת זהות חדש לכל תושב או אזרח ישראלי עם רישום לידתו, או עם עלייתו ארצה.

מספר תעודת זהות מורכב מתשע ספרות. כמובן, לא כל מספר בן תשע ספרות הוא מספר תעודת זהות תקין.

להלן תיאור תהליך בדיקת תקינות של תעודת זהות. לצורך הדוגמה, נבדוק תקינות של תעודת הזהות הבאה: 123456782.

שלב ראשון: נכפיל כל ספרה בתעודת הזהות ב-1 או ב-2 בהתאם למיקום שלה במספר. ספרה שנמצאת במיקום אי זוגי נכפיל ב-1 וספרה שנמצאת במיקום זוגי נכפיל ב-2. לדוגמה:

ספרה בת"ז	1	2	3	4	5	6	7	8	2
הכפלה ב -	1	2	1	2	1	2	1	2	1
תוצאה	1	4	3	8	5	12	7	16	2

שלב שני: נעבור על כל מספר שהתקבל כתוצאה מפעולת ההכפלה ונבדוק אם הוא גדול מ-9. אם כן, נחבר את שתי הספרות שלו. אחרת, נשאיר אותו כפי שהוא. לדוגמה, 16 גדול מ-9 ולכן נחבר את שתי ספרותיו (1+6) ונקבל 7.

שלב שלישי: נסכום את כל המספרים שיצאו בתוצאה.

שלב רביעי: נבדוק האם המספר שהתקבל כתוצאה מהשלב השלישי מתחלק ב-10 ללא שארית. אם כן, מספר תעודת הזהות תקין, אחרת - לא תקין.

דוגמה של כל תהליך בדיקת תקינות תעודת זהות:

ספרה בת"ז	1	2	3	4	5	6	7	8	2
הכפלה ב -	1	2	1	2	1	2	1	2	1
תוצאה	1	4	3	8	5	12	7	16	2
תיקון תוצאה עבור מספרים גדולים מ-9	1	4	3	8	5	3	7	7	2
סכום מספרים	40								

האם 40 מתחלק ב-10 ללא שארית? כן. מספר תעודת הזהות תקין!

מידע נוסף על בדיקת תקינות תעודת הזהות תוכלו למצוא ב**קישור הבא**.

כתבו קוד העונה על הדרישות בסעיפים הבאים:

1. כתבו פונקציה בוליאנית בשם `check_id_valid` המוגדרת כך:

```
1 def check_id_valid(id_number):
```

הפונקציה מקבלת כפרמטר מספר תעודת זהות (מספר שלם) ומחזירה ערך אמת (True) אם הוא תקין, אחרת מחזירה ערך שקר (False).

```
1 print(check_id_valid(123456780))
2 print(check_id_valid(123456782))
```

```
False
```

```
True
```

הנחיות:

- אין להניח שהקלט שמועבר לפונקציה הוא חוקי - בדקו שאכן התקבל מספר שלם המכיל 9 ספרות בדיוק. אם לא, זרקו חריגה מתאימה על ידי שימוש במנגנון החריגות.
- בדיקת תקינות תעודת הזהות תתבצע באופן שתואר קודם לכן (המלצה: היעזרו ב-one liners).
- שימו לב: בתרגיל זה אין צורך לבדוק מקרה קצה של מספר תעודת זהות תקין עם אפסים מובילים. לדוגמה, המספר 009090929 הנו מספר תעודת זהות תקין (לפי כללי הבדיקה לעיל), אך עליכם להתייחס אליו כמו למספר 9090929 (ללא האפסים), שכמובן איננו תקין כי אינו מכיל 9 ספרות.

2. ממשו מחלקה שתייצג איטרטור בשם `IDliterator`.

הוסיפו למחלקה תכונה בשם `_id` המייצגת מספר תעודת זהות בטווח שבין 0 ל- 9999999999 (תכונה זו מאותחלת בעת יצירת מופע איטרטור של המחלקה).

ממשו במחלקה מתודת `__iter__` המחזירה את מופע האיטרטור.

ממשו במחלקה מתודת `__next__` המחזירה בכל פעם את מספר תעודת הזהות התקין הבא בטווח שבין `_id` (כולל) ל-9999999999.

הנחיות:

אין להשתמש בגנרטורים.

3. כדי לבדוק שהקוד שכתבתם עובד, כתבו תוכנית ראשית ובה צרו איטרטור מן המחלקה שיצרתם.

אתחלו את האיטרטור עם ערך תעודת הזהות הבא 123456780, הפיקו באמצעותו 10 מספרים של תעודות זהות חדשות והדפיסו אותם למסך. להלן הפלט שאתם אמורים לקבל:

123456782

123456790

123456808

123456816

123456824

123456832

123456840

123456857

123456865

123456873

שימו לב: אל תשכחו להגדיר את תנאי העצירה לאיטרטור! חשבו מה הוא.

4. כתבו **פונקציית גנרטור** בשם `id_generator` המתנהגת בצורה דומה לאיטרטור `IDIterator`. פונקציית הגנרטור מקבלת כפרמטר מספר תעודת זהות. בכל פעם שמבקשים ממנה להפיק ערך, היא מפיקה את מספר תעודת הזהות התקין הבא בטווח (עד למספר - 999999999).

5. כדי לבדוק שהקוד שכתבתם עובד, כתבו תוכנית ראשית ובה צרו גנרטור באמצעות פונקציית הגנרטור `id_generator`. אתחלו את הגנרטור עם ערך תעודת הזהות הבא 123456780, הפיקו באמצעותו 10 מספרים של תעודות זהות חדשות והדפיסו אותם למסך.