תרגיל 3

Strings manipulations, Dictionaries, and Intro to Text Analysis

תאריך פרסום: 05/11/2020

תאריך הגשה: 22/11/2020

מתרגל אחראי: ישעיה צברי

הנחיות כלליות:

* העבודה תבוצע ביחידים.
* קראו את ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
* מומלץ לקרוא את כל העבודה לפני תחילת הפתרון.
* כתבו תיעוד (הערות) שמסביר את הקוד שלכם. אין לכתוב הערות בעברית.
* עליכם להוריד את הקובץ “hw3.py” מתיקיית "תרגיל בית 3" מהמודל, ולהכניס את הקוד שלכם בשורות המתאימות. תבצעו את השינויים שלכם בהתאם להערות בקובץ.
* אין להשתמש בחבילות או במודולים, אלא אם נאמר במפורש.
* ניתן להניח שהקלט תקין, אלא אם נכתב אחרת בשאלה.
* העבודה תיבדק באופן אוטומטי ולכן על הפלטים להיות בדיוק כפי שמוגדר בתרגיל (ללא רווחים מיותרים בהדפסות).
* העתקת קוד (משנים קודמות, מחברים או מהאינטרנט) עלולה להוביל **לכישלון בקורס. אל תעתיקו!**
* שאלות בנוגע לעבודה ישאלו ב-"פורום שאלות לתרגיל בית 3" במודל או בשעות הקבלה של המתרגל האחראי בלבד.
* את העבודה יש להגיש דרך מערכת ההגשה בכתובת: <https://subsys.ise.bgu.ac.il/submission/login.aspx>
  + יש להתחבר עם שם המשתמש והסיסמא של האוניברסיטה בצירוף תעודת זהות.
  + בכדי להעלות עבודה יש לבחור את הקורס Intro ואת משימה Assignment3.
  + יש להעלאות את הקובץ הרצוי וללחוץ על Upload.
  + העבודה הועלתה בצורה תקינה רק לאחר שמוצגת הודעה "File uploaded successfully".
* טרם ההגשה אנא וודאו:
  + כל אחד מהקבצים מתקמפל ורץ כנדרש.
  + המשתנים שכתבתם עם שמות משמעותיים (ללא שמות כמו a).
  + בתחילת כל שאלה כתבתם הערות לקוד באנגלית.
  + אין הדפסות מיותרות (למשל הרצה של טסטים).
* מטרות העבודה: תרגול עבודה עם מחרוזות (str) ומבני נתונים.
* למידע על built-in types בpython גשו ל:   
  <https://docs.python.org/3/library/stdtypes.html>
* לסיכום על python iterables (רשימות, מילונים וכו׳) גשו ל:  
  <https://www.pythonlikeyoumeanit.com/Module2_EssentialsOfPython/Iterables.html>
* למידע על ערך ascii גשו ל:   
  <https://en.wikipedia.org/wiki/ASCII>
* **תוכן העבודה בחלקים ב׳ וג׳ עוסק בנושאים אותם נתחיל לתרגל בשבוע 4, לכן, מומלץ לפתור עד התרגול הרביעי את חלק א׳ בלבד. אם ביכולתכם לפתור את כל העבודה לפני, אתם בהחלט רשאים.**

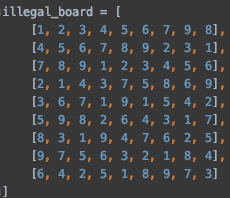
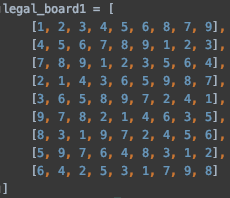
**בהצלחה!**

**חלק א׳ - חימום:**

בחלק זה נוודא תקינות של לוח סודוקו פתור. [להסבר על סודוקו](https://en.wikipedia.org/wiki/Sudoku).   
ממשו פונקציה ***sudoku\_is\_legal(board)*** אשר מחזירה (באמצעות פקודת ה-return) ערך בוליאני True כאשר לוח המשחק הפתור שהתקבל אכן חוקי, ו-False אחרת. הפונקציה מקבלת פרמטר יחיד board, רשימה דו-מימדית (מקוננת) בגודל 9\*9 (תשע שורות עם תשעה תאים כל אחת) המייצגת לוח סודוקו פתור.  
הניחו קלט תקין:

* הרשימה המקוננת board בגודל 9\*9.
* הרשימה מכילה רק את הספרות 1-9.

דוגמאות:  
1. הלוח illegal\_board **לא** חוקי ולכן הקריאה sudoku \_is\_legal(board=illegal\_board) תחזיר False.   
2. הלוח legal\_board1 חוקי ולכן הקריאה sudoku\_is\_legal(board=legal\_board1) תחזיר True.

legal\_board1 = [

[1, 2, 3, 4, 5, 6, 8, 7, 9],

[4, 5, 6, 7, 8, 9, 1, 2, 3],

[7, 8, 9, 1, 2, 3, 5, 6, 4],

[2, 1, 4, 3, 6, 5, 9, 8, 7],

[3, 6, 5, 8, 9, 7, 2, 4, 1],

[9, 7, 8, 2, 1, 4, 6, 3, 5],

[8, 3, 1, 9, 7, 2, 4, 5, 6],

[5, 9, 7, 6, 4, 8, 3, 1, 2],

[6, 4, 2, 5, 3, 1, 7, 9, 8]

]

illegal\_board = [

[1, 2, 3, 4, 5, 6, 7, 9, 8],

[4, 5, 6, 7, 8, 9, 2, 3, 1],

[7, 8, 9, 1, 2, 3, 4, 5, 6],

[2, 1, 4, 3, 7, 5, 8, 6, 9],

[3, 6, 7, 1, 9, 1, 5, 4, 2],

[5, 9, 8, 2, 6, 4, 3, 1, 7],

[8, 3, 1, 9, 4, 7, 6, 2, 5],

[9, 7, 5, 6, 3, 2, 1, 8, 4],

[6, 4, 2, 5, 1, 8, 9, 7, 3]

]

**חלק ב׳ – string manipulations:**

|  |  |
| --- | --- |
| סימן | טון |
| ?! | שלילי |
| ! | שלילי |
| ? | חיובי |
| כל תו אחר | חיובי |

1. נגדיר "טון של משפט" כחיובי או שלילי לפי הסיום שלו. סיום של משפט ב-! או ב-?! מעידים על טון שלילי, ו-? בסיום המשפט, או כל תו אחר, מעידים על טון חיובי.  
   נגדיר את התו ״!״ בסיום משפט כהופכי לתו ״?״ . נגדיר את התת-מחרוזת "?!" בסיום משפט כהופכית לכל תו שונה מ-"!" ו מ-"?" בסיום משפט. שני יחסי ההופכי שהגדרנו הינם יחסים סימטריים. בתרגיל זה נשנה את הסיום של משפט נתון ובכך נשנה או לא נשנה את הטון של המשפט המקורי.  
   ממשו פונקציה ***change\_tone(phrase, new\_tone)*** אשר מחזירה (באמצעות פקודת ה – return) משפט לאחר מניפולציה של הטקסט. הפונקציה מקבלת שני פרמטרים:
   1. phrase – משתנה מסוג str המייצג משפט (הניחו כי המשפט לא ריק).
   2. new\_tone - משתנה בוליאני וערכו True או False.

הפונקציה תחזיר מחרוזת זהה ל - phrase למעט השינויים והתנאים הבאים:

* אם new\_tone is True הפונקציה **תהפוך** את הטון של המשפט ע״י שינוי הסיום, כלומר התו או השניים האחרונים, ללא שינויים נוספים. לדוגמה, בהינתן המשפט:  
   phrase=“You don’t mess with cats!” ו – new\_tone is True   
  הפונקציה תחזיר את המחרוזת:  
  “You don’t mess with cats?” .
* אם new\_tone is False הפונקציה תחליף את הסימן טון הנוכחי בסימן המסמל את אותו הטון.  
  לדוגמה בהינתן המשפט:  
  phrase = “prepare to takeoff!?” ו – new\_tone is False  
  הפונקציה תחזיר את המחרוזת:  
  “prepare to takeoff!”.

הערות:

* רווח הוא תו בפני עצמו, אין למחוק אף רווח שלא לצורך.
* כאשר סימן שאינו ״?״ או ״!״ מומר לסימן ״?!״, הסימן ״?!״ מתווסף ואינו מוחק אף תו אחר.

1. פסקה אחת יכולה להכיל מספר משפטים, כל משפט מסתיים בנקודה (״.״), למעט המשפט **האחרון** שעשוי להסתיים בתו שאינו נקודה.  
   ממשו את הפונקציה be\_polite(paragraph), אשר מחזירה (באמצעות פקודת ה – return) פסקה לאחר מניפולציה של הטקסט. הפונקציה מקבלת פרמטר יחיד paragraph מסוג str המייצג פסקה (הניחו כי הפסקה אינה ריקה).

לדוגמה, בהינתן הקלט:  
paragraph = “I’m not mad!. I want soup?”

הפלט של הפונקציה יהיה משתנה מסוג str שערכו:  
“I’m not mad?. I want soup!”

**חלק ג׳ - python data structures:**

פייתון היא שפת תכנות עם מבני זיכרון מובנים (built-in) רבים ומגוונים. מבני נתונים שונים, שלעיתים נדמים כזהים, ממומשים באופן שונה ומצטיינים בשימוש למטרות שונות. בחלק זה נתרגל שימוש במבנים: list, tuple, dictionary, set. כמו כן, נשתמש בערך ascii לייצוג תווים ע״מ למיין אותם, ייצוג תווים לפי ערך ascii מאפשר הצגה ושימוש בטקסט בין מערכות ותוכנות שונות באופן אחיד.

1. הגדרות:  
   רשימה עם חזרות מוגדרת כרשימה שבה אובייקט מסוים (לדוגמה כיסא) יכול להופיע מספר פעמים גדול מ1. לעומתה, רשימה ללא חזרות מוגדרת כרשימה בה מספר ההופעות המקסימלי של כל אובייקט הוא 1. לדוגמה, הרשימה ׳חתול, כלב, מטוס׳ היא רשימה **ללא** חזרות, לעומת ׳חתול, כלב, חתול, מטוס׳ היא רשימה **עם** חזרות.

ממשו את הפונקציה print\_chars(phrase, repeat) אשר מחזירה (באמצעות פקודת ה – return) את רשימת התווים **ממוינת** שהופיעו במשפט מסוים. הפונקציה מקבלת שני פרמטרים:

1. phrase - מחרוזת המייצגת משפט (ללא קשר להגדרות הסעיף הקודם).
2. repeat - משתנה מסוג bool.

בהינתן repeat is False הפונקציה מחזירה את רשימת התווים המופיעים ב - phrase **ממוינת** **ללא** חזרות. בהינתן repeat is True הפונקציה מחזירה את רשימת התווים המופיעים ב – phrase **ממוינת עם** חזרות.

הערות:

* רווח (space) הוא תו.
* אותיות גדולות וקטנות הן תווים שונים בתכלית ( ‘b’ != ‘B’ ).
* מיון גנרי של מחרוזות בפייתון מתבצע עפ״י הערך הascii שלהם.

דוגמאות:

* בהינתן הקלט phrase = “Bojack you are a crazy horse” ו – repeat is False   
  הפונקציה תחזיר את הרשימה [' ', 'B', 'a', 'c', 'e', 'h', 'j', 'k', 'o', 'r', 's', 'u', 'y', 'z’].
* בהינתן הקלט phrase = “Ada was first” ו – repeat is True   
  הפונקציה תחזיר את הרשימה [' ', ' ', 'A', 'a', 'a', 'd', 'f', 'i', 'r', 's', 's', 't', 'w'] .

1. במכללת TheyCode מערכת ההרשמה תקולה ואינה מתפקדת היטב.
   1. בהרשמה ליום הפתוח במחלקה לחקר חייזרים, כל מועמד יכול להירשם מס׳ פעמים עם אותו שם מלא וציון פסיכומטרי. המכללה דורשת שכל מועמד יופיע פעם יחידה ברשימת הנרשמים ובאופן תקין, כך שהשם הפרטי ושם המשפחה של הנרשם יחלו באות גדולה (capital letter) ושאר האותיות קטנות, כמו כן, על הציון המוזן להיות בטווח תקין (200-800).  
      ממשו את הפונקציה orientation\_day\_registery(all\_listed\_students) אשר מחזירה (באמצעות פקודת ה – return) מבנה נתונים המאחסן את ציוני המועמדים לצד השם המלא שלהם. הפונקציה מקבלת פרמטר יחיד all\_listed\_students, רשימה של מחרוזות (str). כל מחרוזת מייצגת מועמד וציון במבנה מוגדר.  
      הערות:
      1. מבנה כל מחרוזת המייצגת מועמד הוא “FirstName LastName SAT” (SAT הוא שמו של המבחן האמריקאי המקביל לפסיכומטרי), כאשר בין שם פרטי לשם משפחה, ובין שם משפחה לציון, רווח בודד וללא שמות אמצעיים.
      2. **שימו לב**, עליכם לאכוף את תקינות השמות עפ״י ההגדרה הנ״ל.
      3. עליכם לשמור את הציון של כל מועמד כמשתנה מסוג int.
      4. בהינתן ציון לא תקין (כשערכו לא בטווח) הפונקציה תשמור למועמד את הציון 200 במקומו.
      5. על מבנה הנתונים לאפשר שליפה של ציון מועמד בהינתן שמו המלא ובמהירות הגבוהה ביותר האפשרית.
      6. עקב האפשרות למועמד להירשם מס׳ פעמים, הציון **האחרון** שהזין הוא הציון הקובע.

לדוגמה, בהינתן הקלט:  
all\_listed\_students = [“Katherine Johnson 20”, “Marie CUrie 790”, “Katherine johHson 800”, “Margaret Hamilton 30”]   
הפונקציה תחזיר מבנה נתונים מובנה (built-in) של פייתון, **לדוגמה**:  
{“Katherine Johnson”: 800,  
 “Marie Curie”: 790,   
“Margaret Hamilton”: 200}

* 1. לאחר ההרשמה, המכללה מעוניינת לאתר את המועמדים המצטיינים, ע״מ לשלוח להם מידע על תוכניות מצטיינים ומלגות.
     1. ממשו את הפונקציה get\_number\_of\_honors(students) אשר מחזירה (באמצעות פקודת ה – return) את מספר המועמדים שציון הפסיכומטרי שלהם גדול או שווה ל750. הפונקציה מקבלת פרמטר יחיד students, שהוא מבנה נתונים מהסוג שבחרתם בסעיף a המכיל רישום של מועמדים וציון הפסיכומטרי שלהם.  
        לדוגמה, בהינתן מבנה נתונים המכיל מועמד א׳ עם ציון 700 ומועמד ב׳ עם ציון 750, הפונקציה תחזיר משתנה מסוג int שערכו 1.
     2. ממשו את הפונקציה get\_honors(students, min\_grade) אשר מחזירה (באמצעות פקודת ה – return) את רשימה ממוינת של שמות המועמדים שציון הפסיכומטרי שלהם גבוה או שווה לערך המטרה.   
        הפונקציה מקבלת שני פרמטרים:
        1. students – שהוא מבנה נתונים מהסוג שבחרתם בסעיף a המכיל רישום של מועמדים וציון הפסיכומטרי שלהם.
        2. min\_grade – ערך המטרה, משתנה מסוג int בעל ערך בין 200 ל-800.

הערה, מיון הרשימה הוא עפ״י ציון הפסיכומטרי של המועמדים **מהגבוה לנמוך**, משמע, שם המועמד בעל ציון הפסיכומטרי הגבוה ביותר (בהנחה שהוא גבוה מmin\_grade - ) יופיע בתא הראשון ברשימה.

לדוגמה, בהינתן מבנה נתונים students המכיל מועמדת א׳ בשם Ada Lovelace עם ציון 750, מועמדת ב׳ בשם Marie Curie עם ציון 730 ומועמדת ג׳ בשם Katherine Johnson עם ציון 710, ומשתנה מטרה min\_grade = 730, הפונקציה תחזיר את הרשימה [“Ada Lovelace”, “Marie Curie”].

* + 1. ממשו את הפונקציה get\_with\_honors\_by\_avg(students\_list) אשר מחזירה (באמצעות פקודת ה – return) משתנה מסוג tuple המכיל את ממוצע ציוני הפסיכומטרי של כל המועמדים ואת רשימת המועמדים הממוינת שציון הפסיכומטרי שלהם גדול או שווה לממוצע זה. הפונקציה מקבלת פרמטר יחיד, students\_list, רשימה המכילה מחרוזות של שמות המועמדים וציון הפסיכומטרי שלהם, במבנה **זהה** למתואר בסעיף a.   
       הערות:
       1. הרשימה students\_list היא הרשימה הגולמית בדומה לסעיף א׳, לכן ייתכנו כפילויות של שמות המועמדים וציונים לא תקינים, עליכם לאכוף את תקינות השמות והציונים בדומה לסעיף א׳.
       2. עקב האפשרות למועמד להירשם מס׳ פעמים, הציון האחרון שהזין הוא הציון הקובע.
       3. מיון רשימת המועמדים המצטיינים הוא עפ״י ציון הפסיכומטרי שלהם, מהגבוה לנמוך (בדומה לסעיף הקודם).

לדוגמה, בהינתן הקלט:  
students\_list=[“Rosalind Franklin 795”, “Albert Einstein 760”,  
 “Bill Nye 420”, “Ada Lovelace 50”]   
הפונקציה תחזיר tuple עם הערכים:  
(543.75, [“Rosalind Franklin”, “Albert Einstein”]).

**העשרה (ללא ניקוד)**: חידה, מה משותף לכלל השמות המופיעים בדוגמאות ההרצה בחלק זה?