

## תרגיל 2 – פונקציות, לולאות, רשימות

תאריך פרסום: 29/10/2018

תאריך הגשה: 5/11 בשעה 23:59

מתרגל אחראי: שגיא תובל

### הנחיות כלליות:

- קראו את כל ההוראות לגבי הגשת תרגילי הבית באתר הקורס.
- קראו את כל העבודה לפני שתתחילו לפתור אותה.
- כתבו תיעוד (הערות) שמסביר את הקוד שלכם. אין לכתוב הערות בעברית.
- שאלות בנוגע לעבודה יישאלו ב-"פורום שאלות לתרגיל בית 2" במודל או בשעות הקבלה של המתרגל האחראי בלבד.
- כתבו את כל הקוד שלכם בקובץ 'hw2.py'. אין צורך לבצע הפרדה בין התרגילים, רק לממש את הפונקציות הנדרשות.
- השימוש בחבילות חיצוניות אסור בהחלט ויגרור ציון 0.
- על הפלטים להיות בדיוק כפי שמוגדר בתרגיל (ללא רווחים מיותרים בפלטי טקסט print).
- יש להקפיד על חתימות זהות של הפונקציות, שכן מערכת הבדיקה קוראת לפונקציות בצורה אוטומטית.

### תרגיל 1:

ממשו את הפונקציה הבאה:

`def print_digits_frame(digit)` - הפונקציה מקבלת ספרה ומדפיסה `digit + 2` שורות באופן הבא:

1. השורות הראשונה (0) והאחרונה (`digit + 1`) תהיינה מורכבות מ-`digit + 2` סולמיות (#) רציפות.
2. כל שורה אחרת תכיל את כל הספרות עד מספר השורה, ותיתחם בתו המאונך (|) בתחילתה ובסופה. שימו לב כי יש להכניס תווי רווח (' ') כדי ליצור מסגרת תקינה.

ניתן להניח כי הקלט תקין (לא יינתן קלט שאינו מספר).

דוגמאות:

In: `print_digits_frame(7)`

Out:

```
#####
| 1      |
| 12     |
| 123    |
| 1234   |
| 12345  |
| 123456 |
| 1234567|
#####
```

**In:** print\_digits\_frame(3)

**Out:**

```
#####
| 1 |
| 12 |
| 123 |
#####
```

תזכורת: ניתן לשרשר מחרוזות (strings) באמצעות פעולות חיבור (+) וכפל (\*). לדוגמא:

**In [6]:** 'Good ' + 'luck' + ('!' \* 3)

**Out[6]:** 'Good luck!!!'

## תרגיל 2:

"דחיסת נתונים היא התחום העוסק בהקטנת נפחם של נתונים, כך שיתפסו פחות מקום מאשר בצורתם המקורית." ([ויקיפדיה – דחיסת נתונים](#)).

בשאלה זו נממש דחיסת נתונים פשוטה של מחרוזת ונבחן את איכות הדחיסה. ממשו את שתי הפונקציות הבאות:

**def compress\_text(to\_compress)** – הפונקציה מקבלת מחרוזת, ממירה אותה לייצוג דחוס על ידי מנייה של אותיות החוזרות על עצמן ברצף. הגרסה הדחוסה תיוצג באמצעות רשימה, שתוחזר לקורא הפונקציה, ותמומש באופן הבא:

- כל איבר ברשימה יהיה רשימה אחרת המכילה שני איברים - התו הנצפה ומספר ההופעות שלו ברצף.
- האיברים יהיו מסודרים לפי הסדר שהתקבל מהמשתמש.

בכך, במקום לשמור את אותו מספר פעמים, נשמור רק את מספר ההופעות של התו.

דוגמאות קלט-פלט של הפונקציה:

to_compress	Return value
aaaaabbbccccc	[['a', 5], ['b', 3], ['c', 4]]
aabbcc	[['a', 2], ['b', 2], ['c', 2]]
abcd	[['a', 1], ['b', 1], ['c', 1], ['d', 1]]

**def evaluate\_compression(original\_text, compressed\_text)** – הפונקציה מקבלת מחרוזת ואת המחרוזת הדחוסה המתאימה לה (שהוחזרה למשתמש מהפונקציה הראשונה) ומחשבת את יחס הדחיסה, כלומר

את היחס בין מספר האיברים (תווים או מספרים) בין הייצוג המקורי לדחוס. יחס הדחיסה יהיה הפלט שיוחזר. כך לדוגמא, אם המחרוזת המקורית מכילה 16 תווים והייצוג הדחוס דורש 4 איברים (שני תווים ושני מספרים), יחס הדחיסה יהיה 4.

דוגמאות קלט-פלט:

original text	compressed text	Return value
aaaaabbccccc	[[ 'a', 5], [ 'b', 3], [ 'c', 4]]	2.0
aabbcc	[[ 'a', 2], [ 'b', 2], [ 'c', 2]]	1.0
abcd	[[ 'a', 1], [ 'b', 1], [ 'c', 1], [ 'd', 1]]	0.5

### תרגיל 3:

נכתוב תכנית המחשבת את ממוצע הציונים של סטודנטים ומאפשרת מתן משוב מתאים.

ממשו את שתי הפונקציות הבאות:

**def calculate\_average\_grade(students, feedback)** – הפונקציה מקבלת רשימה דו מימדית של tuples ומשתנה בוליאני feedback.

כל tuple ברשימה students מכיל שני איברים, שם (string) וציון (int), לדוגמא:

```
students = [('David', 100), ('Ben', 90), ('Gurion', 90), ('Tau', 60)]
```

הפונקציה מחשבת ומדפיסה את ממוצע הציונים של הסטודנטים (אינה מחזירה פלט):

“Average grade: GRADE”

בנוסף, הפונקציה מאפשרת הדפסת משוב לכל סטודנט - כאשר ערך המשתנה feedback הוא True, הפונקציה תקרא לפונקציה אחרת בשם print\_feedback שתדפיס משוב מתאים לכל סטודנט בהתאם לציון שלו ביחס לממוצע:

**def print\_feedbacks(students, average\_grade)** – הפונקציה מקבלת כקלט את רשימת הסטודנטים וציוניהם וכן את הציון הממוצע ומדפיסה משוב לכל סטודנט. אם הציון של הסטודנט גדול או שווה לממוצע, יודפס: “Good job, NAME!” אחרת: “You can do better, NAME!”

דוגמאות:

**In:** calculate\_average\_grade( ('David', 100), ('Ben', 90), ('Gurion', 90), ('Tau', 60)), True)

**Out:**

```
Average grade: 85.0
Good job, David!
Good job, Ben!
Good job, Gurion!
You can do better, Tau!
```

**In:** `calculate_average_grade( ('David', 100), ('Ben', 90), ('Gurion', 90), ('Tau', 60)), False)`

**Out:**

Average grade: 85.0

### הנחיות הגשה:

- יש להגיש את העבודה למערכת ההגשה כפי שמתואר בהנחיות ההגשה במודל.
- יש להגיש קובץ מכוון אחד בלבד בשם 'hw2.zip' (לא rar או שום סיומת אחרת), כאשר בתוכו נמצאת תיקייה בשם hw2 ובתוכה הקובץ 'hw2.py'. כל מבנה, שם אחר או סיומת אחרת ייגררו ציון 0.

😊 בהצלחה!