

תרגיל בית מספר 2 - להגשה עד 26 בנובמבר בשעה 23:55

בתרגיל זה ניתן להגיע למקסימום של 100 נקודות.

- הקפידו לענות על כל מה שנשאלתם. בכל מקום בו השתמשתם בפיתון רישמו את הפקודות שביצעתם.
- תשובות מילוליות והסברים צריכים להיות תמציתיים, קולעים וברורים. להנחיה זו מטרה כפולה:
 1. על מנת שנוכל לבדוק את התרגילים שלכם בזמן סביר.
 2. כדי להרגיל אתכם להבעת טיעונים באופן מתומצת ויעיל, ללא פרטים חסרים מצד אחד אך ללא עודף בלתי הכרחי מצד שני. זוהי פרקטיקה חשובה במדעים המדוייקים ובהנדסה.

הנחיות הגשה והערות:

- ההגשה תתבצע גם ב- moodle וגם עותק מודפס בתרגול או בהרצאה באופן הבא:
 - יחד עם התרגיל פורסם קובץ template ובו כותרות הפונקציות השונות שעליכם לממש. ממשו את הפונקציות הדרושות במקומות המיועדים.
 - קובץ זה הגישו **במודל**: עבור סטודנטית שמספר ת"ז שלה הוא 012345678, שנו את שם הקובץ template.py ל- 012345678.py.
 - **בנוסף**, יש להגיש במודל קובץ pdf עבור השאלות המילוליות וקטעי הקוד אותם מימשתם: עבור סטודנטית שמספר ת"ז שלה הוא 012345678, שם הקובץ יהיה 012345678.pdf.
 - **סה"כ יש להגיש 2 קבצים לתיבה המתאימה במודל**
- קובץ templatn:
 - מצורף לתרגיל קובץ שלד לפתרון, המכיל את כותרות הפונקציות שעליכם להשלים.
 - **שימו לב!** הקובץ עצמו לא עובד (בהרצה שלו תתקבל שגיאה), עליכם להשלים את הפונקציות כך שלא תתקבלנה שגיאות.
 - לנוחיותכם- מצורפת פונקציית test() לקובץ, **אין צורך להבין איך היא עובדת**, אולם אם לא תהיינה שגיאות, כאשר תריצו את קובץ templatn היא תדפיס לבסוף test done!, אחרת היא תדפיס באיזו פונקציה ישנה שגיאה
 - התרגיל עצמו ייבדק על קריאות שונות מהפונקציה המצורפת!

איחור בהגשה:

לכל סטודנט יש 5 ימי איחור מצטברים לכל הסמסטר. כלומר ניתן למשל להגיש את תרגיל 2 באיחור של יומיים, את תרגיל 3 באיחור של יום, ואת תרגיל 5 באיחור של יומיים. כל איחור, אפילו של דקה משעת ההגשה, נחשב כיום איחור. אין צורך לבקש להגיש באיחור, כל עוד לא חרגתם ממכסת 5 הימים. איחורים מעבר ל- 5 ימים אלו לא יתקבלו.

אוניברסיטת תל אביב - בית הספר למדעי המחשב

מבוא כללי לתכנות ולמדעי המחשב, מקבץ הסייבר, חורף 2017

שאלה 1 (10 נק') – פונקציות "ספריה" בפייתון

ראינו בכיתה מספר פונקציות מובנות (built-in) של פייתון, כדוגמת print, input, type וכו'. ישנן פונקציות רבות נוספות, שחלקן מסודרות בתוך ספריות (libraries). ספריות שונות מכילות פונקציות שקשורות לתחומים ולשימושים שונים. בתרגיל זה תכירו שתי ספריות חשובות. כדי להשתמש בפונקציית ספריה, יש לרשום בתחילת הקובץ פקודה שמצהירה על השימוש באותה ספריה. הפקודה היא `import <lib_name>` כאשר `<lib_name>` הוא שם הספריה. למשל עבור ספריה בשם `time`, המכילה פונקציות שונות שקשורות לחישובי זמנים, נרשום:

```
import time
```

לאחר מכן נוכל להשתמש בכל הפונקציות מאותה ספריה, באופן הבא: `<lib_name>.<func_name>` כאשר `<func_name>` הוא שם הפונקציה המבוקשת. למשל, בספריה `time` ישנה פונקציה בשם `ctime()`, שמדפיסה את הזמן הנוכחי. כדי לקרוא לה נרשום:

```
time.ctime()
```

שימו לב שהפונקציה `ctime` לא מקבלת שום פרמטרים, לכן הסוגריים ריקים.

הספריה `time`

א. בשיעור ראינו 3 גרסאות לפונקציה `sum_range`, כעת נרצה להשוות את יעילותן.

```
import time
t0 = time.clock()
res = sum_range3(1000)
t1 = time.clock()
print(t1-t0)
```

הפונקציה `clock()` מהספריה `time` לא מקבלת שום פרמטר ומחזירה את כמות השניות שעברו מאז הפעם הקודמת בה קראו לה. למשל, הקוד כאן מודד את זמני הריצה של `sum_range3(n)` עבור `n=1000`

הקוד הזה מודד את זמן הריצה של הקוד שבין שתי פקודות

ה- `time.clock()`. עבור שלוש הגרסאות של `sum_range(n)` מהשיעור השלישי, מדדו את זמני

הריצה על הקלטים `10**5`, `10**6`, `10**7`, ורשמו את התוצאות בטבלה.

איזו גרסה היא היעילה ביותר מבין השלוש? פי כמה לדעתכם יגדל זמן הריצה עבור כל אחת

מהפונקציות אם נגדיל את הקלט פי 2?

להעשרה: רשימת הספריות של פייתון 3 <http://docs.python.org/release/3.2/library/index.html>

אוניברסיטת תל אביב - בית הספר למדעי המחשב
מבוא כללי לתכנות ולמדעי המחשב, מקבץ הסייבר, חורף 2017

שאלה 2 (35 נק') – לולאות

כתבו פונקציה המממשת את המשחק "7 בום!"
הפונקציה תקבל ערך מספרי בודד :

```
def boom(n):
```

```
    ...
```

ותדפיס את כל הערכים מ 1 ועד n (כולל) למעט במקרים הבאים :

א. אם הערך מתחלק ב-7 יש להדפיס "boom!"

ב. אם הספרה 7 מופיעה בערך, יש להדפיס "boom!"

ג. אם הערך מתחלק ב7 וגם מופיעה בו הספרה 7 יש להדפיס "boom! boom!"

למשל- עבור הערך 17 יש להדפיס "boom!", עבור הערך 77 יש להדפיס "boom! boom!", עבור הערך 3 יש להדפיס 3

** שימור לב שהפונקציה לא מחזירה דבר.
להלן דוגמת הרצה :

```
>>>boom(7)
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
"boom! boom!"
```

שאלה 3 (20 נק') – חיפוש בינארי

השאלה עוסקת באלגוריתם לחיפוש בינארי ברשימה ממוינת (יילמד בשיעור מספר 4).

א. נתונה הרשימה הבאה: $L = [1,2,3,4,5,6,7,8,9,10,11,12]$. מבצעים בה חיפוש בינארי למציאת איבר x כלשהו, וידוע שלאחר 4 איטרציות האלגוריתם מחזיר True. כלומר, האלגוריתם בדק בסה"כ 4 פעמים את התנאי

```
if my_list[mid]==x:
```

כאשר ב-3 הפעמים הראשונות התנאי היה False והאלגוריתם המשיך בחיפוש, ואילו בפעם הרביעית

התנאי היה True ואלגוריתם עצר (עם return True).

רישמו את כל הערכים האפשריים עבור x והסבירו בקצרה.

ב. האם יכול להיות שהאלגוריתם יבצע יותר מ-4 איטרציות? אם כן, ציינו דוגמה לערך x אותו מחפשים ברשימה הנ"ל כך שמתקיים מצב זה וציינו כמה איטרציות יתבצעו. אחרת, הסבירו מדוע לא.

אוניברסיטת תל אביב - בית הספר למדעי המחשב
מבוא כללי לתכנות ולמדעי המחשב, מקבץ הסייבר, חורף 2017

שאלה 4 (25 נק') – רשימות דו מימדיות

ראינו בכיתה כי רשימות בפייתון יכולות להכיל ערכים מכל סוגי המשתנים (למשל [1,"2",3.0] הינה רשימה תקנית בפייתון). רשימות יכולות גם להכיל רשימות נוספות בתור ערכים, רשימה שאיבריה הם רשימות נקראת גם רשימה דו מימדית.

****ברשימה דו מימדית כדי להגיע לערך בתוך רשימה פנימית יש צורך ב2 ערכי אינדקס****

לדוגמא : את המטריצה* $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ נוכל לייצג בפייתון באמצעות הרשימה הדו מימדית הבאה:

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

* מטריצה הנה מערך דו-מימדי של מספרים. אפשר לגשת אל כל מספר במטריצה באמצעות מספר השורה ומספר הטור בו הוא נמצא.

א. הריצו את הפקודות הבאות, וציינו מהי התוצאה

```
>>> lst2d = [[1,2,3], [4,5,6,7], [8,9,10]]
>>> lst2d [0][1]
_____
>>> len(lst2d)
_____
>>> len(lst2d [1])
_____
>>> lst2d [2][2]
_____
>>> lst2d [0]
```

ב. כתבו פונקציה בשם lst2d_contains, המקבלת רשימה דו-מימדית וערך נוסף val :
def lst2d_contains (lst2d, val):
הפונקציה תעבור על כל הערכים mat ותחזיר True אם val נמצא בתוך mat, False אחרת.

דוגמאות הרצה :

```
>>> lst2d_contains ([ ['a','b'], [1,2,3] ], 2)
True
>>> lst2d_contains ([ ['a','b'], [1,2,3] ], 10)
False
>>> lst2d_contains ([ ['a','b'], [1,2,3] ], 'b')
True
>>> lst2d_contains ([ ['a','b'], [1,2,3] ], 'aa')
False
```

*בתרגיל זה יש להשתמש בלולאות מקוננות (לולאה בתוך לולאה), לנוחיותכם שלד לפתרון נמצא בקובץ ה template המצורף. השלד מכיל כבר את כותרות 2 לולאות for, עליכם להשלים את שאר הפקודות בפונקציה.

אוניברסיטת תל אביב - בית הספר למדעי המחשב
מבוא כללי לתכנות ולמדעי המחשב, מקבץ הסייבר, חורף 2017

שאלה 5 (10 נק') – זיהוי טעויות בקוד

אמיר ביקש מדין שיכתוב לו פונקציה המקבלת רשימה של מספרים ומחזירה True אם כל המספרים ברשימה זוגיים, אחרת היא תחזיר false. דין כתב את הפונקציה מאוחר בלילה:

```
def all_even(lst):  
    for i in range(len(lst)):  
        if i%2!=0:  
            return False  
    return True
```

כאשר הריץ אמיר את הפונקציה גילה כי היא לא עובדת כמצופה. מצאו את הטעויות של דין ותקנו אותן כך שהפונקציה תעבוד כנדרש.
**שימו לב- אין להוריד ולהוסיף שורות. נסו תחילה להבין למה הפונקציה לא עובדת ותקנו את הטעויות בתוך הקוד עצמו.
לנוחיותכם, הפונקציה נמצאת בקובץ הפייתון המצורף לתרגיל.

סוף.