

## תרגיל בית מספר 1 - להגשה עד 22 במרץ בשעה 23:55

**קיראו בעיון את הנחיות העבודה וההגשה המופיעות באתר הקורס, תחת התיקיה assignments. חריגה מההנחיות תגרוור ירידת ציון / פסילת התרגיל.**

הנחיות והערות ספציפיות לתרגיל זה :

- אפשר וכדאי להתחיל לעבוד על התרגיל כבר בשבוע הראשון לסמסטר, לאחר ההרצאה + התרגול הראשונים.  
הגשה :
- תשובות לשאלות 1, 2, 4 ו-6 יש להגיש בקובץ pdf יחיד. **אין לסרוק תשובות מילוליות ולצרפן לקובץ ה pdf.**
- קוד משאלות 3, 5 ו-6 יש לממש בקובץ השלד (skeleton1.py) המצורף לתרגיל זה. אין לצרף לקובץ ה-py את הקוד ששימש לפתרון יתר השאלות.  
**לא לשכוח לשנות את שם הקובץ למספר ת"ז שלכם לפני ההגשה, עם סיומת py.**
- בסה"כ מגישים שני קבצים בלבד. עבור סטודנטית שמספר ת"ז שלה הוא 012345678 הקבצים שיש להגיש הם 012345678.py ו- 012345678.pdf.
- הקפידו לענות על כל מה שנשאלתם.
- 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 במדעי המחשב.

## שאלה 1

להלן תוכנית קצרה בשפת Python:

```
res = 0
num = int(input("Enter a positive integer: "))
while num > 0:
    res = res + (num % 10)
    num = num // 10
print(res)
```

**הסבר קצר:** תפקיד הפונקציה `input()` הוא לקלוט ממשתמש התוכנית ערכים (קלט) הנחוצים להמשך ביצוע התוכנית. הביטוי בתוך הסוגריים `"Enter a positive integer: "` הוא מחרוזת שתודפס למשתמש לפני שיידרש להכניס קלט. הפונקציה `input()` קולטת את הערך שהכניס המשתמש כמחרוזת. ערך זה עובר המרה מטיפוס מחרוזת לטיפוס מספר שלם (ע"י הפעלת הפונקציה `int()` שראינו בתרגול), ואז המשתנה `num` מקבל ערך זה.

- מה תדפיס התוכנית עבור הקלט 2015?
- באופן כללי, מה מדפיסה התוכנית בהינתן לה מספר שלם חיובי כלשהו `n`?
- כאשר רוצים לתת לתוכנית קלט כביטוי אריתמטי, כמו למשל `3*222 + 47`, מתעוררת בעייה כלשהי. נסו להריץ, ציינו מהי בדיוק הפקודה שנכשלה וגרמה לבעייה, והסבירו בקצרה מה הבעייה.
- הפונקציה `eval` של פייתון יכולה לסייע בקלות לפתרון הבעייה הנ"ל. נסו להבין מה עושה הפונקציה (מומלץ ע"י ניסוי וטעייה וקצת חיפוש). הפונקציה מקבלת בסוגריים פרמטר אחד מטיפוס מחרוזת, והיעזרו בה כדי להריץ את התוכנית הנ"ל על `3*222 + 47`. מה מודפס?
- מה קורה כאשר משנים את התנאי `while num > 0` ל- `while num >= 0`? הסבירו בקצרה.

## שאלה 2

כפי שראיתם בהרצאה, ישנן בפייתון פונקציות שמשויכות למחלקה מסויימת, למשל למחלקת המחרוזות (str). באינטרפרטר IDLE, אם תכתבו "str." ותלחצו על המקש tab, תיפתח חלונית עם מגוון פונקציות המשויכות למחלקת המחרוזות. כמובן, אפשר למצוא תיעוד רב על פונקציות אלו ואחרות ברשת. כמו כן אפשר להשתמש בפונקציה help של פייתון. למשל הפקודה help(str.title) תציג הסבר קצר על הפונקציה str.title שראיתם בהרצאה.

### הערה כללית:

פונקציות של מחלקות ניתן להפעיל בשני אופנים שקולים. אם נסמן ב-C את שם המחלקה (למשל המחלקה str), וב-c\_obj אובייקט קונקרטי מהמחלקה C (למשל מחרוזת "abc"), אז שתי הדרכים הן:

(1) C.func(c\_obj,...) , כלומר הפרמטר הראשון הוא c\_obj ואחריו יתר פרמטרים, אם דרושים.

(2) c\_obj.func(...) , כלומר האובייקט c\_obj לא מופיע בתוך הסוגריים אלא לפני שם הפונקציה.

להלן הדגמה על המחלקה str:

```
>>> course_name = "introduction to computer science"
>>> str.title(course_name)
'Introduction To Computer Science'
>>> course_name.title()
'Introduction To Computer Science'
```

בשאלה זו תתבקשו להריץ מספר פונקציות של מספר מחלקות, ולחקור אותן מעט. הסבירו במילים שלכם, במשפט או שניים לכל היותר, מה כל פונקציה מחזירה, ומה תפקיד כל אחד מהפרמטרים שניתנים לה. צרפו לתשובה העתק של דוגמה או שתיים להרצות שביצעתם, המראות כיצד הגעתם לתשובה.

### המחלקה str

- א. הפונקציה isalnum.
- ב. הפונקציה split. למשל:

```
"Intro2ComputerScience".split("c", 1)
```

הפרמטר האחרון הוא אופציונלי. ציינו כיצד מתנהגת הפונקציה אם לא מעבירים לה את הפרמטר האחרון.

- ג. הפונקציה replace.
- ד. הפונקציה join.
- ה. הפונקציה count.
- ו. הפונקציות find ו-index (מה ההבדל ביניהן?).

### המחלקה list

- ז. הפונקציות הבאות: index, count, find, sort, insert, remove, pop, extend, append.

### שאלה 3

במשימה זו תכירו פעולות בסיסיות על קבצים, כגון פתיחת קובץ, סגירת קובץ, קריאה וכתיבה לקובץ. היעזרו בתיעוד שמופיע בקישור הבא (בסעיף 7.2): <https://docs.python.org/3/tutorial/inputoutput.html>. מצורף לתרגיל קובץ טקסט tom\_sawyer.txt שמכיל את תוכן הספר "Adventures of Tom Sawyer" מאת מארק טוויין. שימרו קובץ זה בשם our\_input.txt, באותה תיקיה בה שמרתם את קובץ השלד skeleton1.py. השלימו את הקוד בקובץ השלד כך שיצור קובץ חדש באותה התיקיה בשם output.txt שבו כל שורה מכילה את מספר המילים בשורה המתאימה בקובץ our\_input.txt.

רמז: היעזרו בפונקציה split מסעיף ב' בשאלה הקודמת.

לדוגמא, עבור הקובץ tom\_sawyer.txt, 5 השורות הראשונות של output.txt יהיו:

5  
0  
2  
0  
2

כמו כן, 5 השורות האחרונות של output.txt יהיו:

13  
14  
15  
15  
6

הקוד שלכם אמור לעבוד עבור כל קובץ טקסט שהוא בשם our\_input.txt.

#### שאלה 4

נדון בבעייה החישובית הבאה : בהינתן מספר שלם חיובי  $num$ , נרצה לדעת כמה פעמים מופיעה בו הספרה 0. למשל עבור הקלט 10030 הפלט המתאים הוא 3.

הקלט יינתן באמצעות הפקודה `input` :

```
num = int(input("Please enter a positive integer: "))
```

(לאחר ביצוע פקודה זו, המשתנה `num` יכיל את המספר אותו הכניס המשתמש).

מטרתנו בשאלה היא להשוות את זמני הריצה של שלושה פתרונות אפשריים לבעייה זו (הערה : אנו נדון בבעייה הנ"ל ובשלושת הפתרונות הללו גם בתרגול הראשון/שני, אבל אפשר להתחיל לפתור את השאלה כבר לאחר התרגול הראשון) :

פתרון ראשון :

```
#1st solution
m = num
cnt = 0
while m>0:
    if m%10 == 0:
        cnt = cnt+1
    m = m//10
```

פתרון שני :

```
#2nd solution
cnt = 0
snum = str(num)
for digit in snum:
    if digit == "0":
        cnt = cnt+1
```

פתרון שלישי :

```
#3rd solution
cnt = str.count(str(num), "0")
```

בשלושת הפתרונות הפלט הרצוי יימצא לבסוף במשתנה `cnt` :

```
print(num, "has", cnt, "zeros")
```

(המשך השאלה בעמוד הבא)

**אוניברסיטת תל אביב - בית הספר למדעי המחשב**  
**מבוא מורחב למדעי המחשב, אביב 2015**

כדי למדוד זמן ריצה של פקודה או סדרת פקודות, נשתמש במעין "סטופר":

```
import time
t0 = time.clock()
t1 = time.clock()
print("Running time: ", t1-t0, "sec")
```

- נוסף בראש התוכנית שלנו את הפקודה `import time`
- נוסף מייד לפני קטע הקוד שאת זמן הריצה שלו ברצוננו למדוד את הפקודה: `t0 = time.clock()`
- נוסף מייד לאחר קטע הקוד הנ"ל את הפקודה: `t1 = time.clock()`
- זמן הריצה של קטע הקוד הוא ההפרש `t1-t0`. נוח להציגו למשל כך: `print("Running time: ", t1-t0, "sec")`

הסבר קצר: `time` היא מחלקה של פייתון המאפשרת ביצוע פקודות שונות הקשורות לזמנים. הפקודה `import time` הכרחית על מנת להשתמש במחלקה (היא "מיבאת" אותה). ניתקל במהלך הקורס בדוגמאות רבות ל"יבוא" של מחלקות). הפונקציה `clock` של המחלקה מחזירה את הזמן שחלף מאז הפעלתה בפעם הראשונה. (למידע נוסף: <http://docs.python.org/release/1.5.1/lib/module-time.html>)

א. מדדו את זמן הריצה של 2 הפתרונות הראשונים עבור המספרים: `2**200`, `2**400`, `2**800`, `2**1600`. ציינו מה היו זמני הריצה (אפשר להציג זאת בגרף), והסבירו בקצרה את התוצאות (התייחסו לקצב הגידול כתלות בגודל הקלט).

שימו לב: כדי לנטרל השפעות של פקודות שקשורות להשגת הקלט והצגת הפלט, ואינן חלק מהפתרון עצמו, זמן הריצה לא יכלול את שורת ה- `input` בהתחלה והדפסת הפלט בסוף.

ב. פונקציות מובנות של פייתון, כמו למשל `str.count`, ממומשות בד"כ באופן יעיל למדי, לעיתים אף באמצעות אלגוריתמים מסובכים יחסית. חיזרו על סעיף א' עבור הפתרון השלישי. מבלי להיכנס לפרטי המימוש של `str.count`, האם היא אכן יעילה יותר מבחינת זמן ריצה, בהשוואה לשני הפתרונות הראשונים?

ג. עבור קלטים בעלי מספר ספרות דומה, האם יש לפלט עצמו, כלומר למספר האפסים בקלט, השפעה כלשהי על זמן הריצה של כל אחד מהפתרונות? ביחרו קלטים מתאימים לבדיקת הסוגיה, ציינו מהם הקלטים בהם השתמשתם, הראו את תוצאות המדידות, והסבירו מה היא מסקנתכם.

ד. להלן לולאה פשוטה:

```
num = 2**100
cnt=0
for i in range(num):
    cnt = cnt+1
```

תנו הערכה גסה לזמן שיקח ללולאה להסתיים. ציינו כל הנחה עליה התבססתם בהערכתכם. איך אתם מסבירים זאת, לאור העובדה שבסעיף א' לולאת ה- `for` של הפתרון השני רצה בזמן קצר באופן משמעותי?

## שאלה 5

במשחק הילדים הנודע "7 בום!" המשתתפים צריכים לנקוב במספרים טבעיים בסדר עולה, אך בכל פעם שמגיעים למספר שמתחלק ב-7 או שמופיעה בו הספרה 7, יש לצעוק במקום "boom!". כמו כן, אם מספר גם מתחלק ב-7 וגם מכיל את הספרה 7 (למשל המספר 7) יש לצעוק "boom-boom!".

הגירסה המוכללת של המשחק (שהולכת ונהיית פופולרית יותר ויותר במקום נידח כלשהו) קרויה "k בום!". והגדרתה זהה לנ"ל כאשר k הינו מספר שלם כלשהו בין 1 ל-9 כולל.

כיתבו תוכנית בשפת פייתון, שמדפיסה את כל המספרים בין 1 ל- $n$  (כולל) בהתאם לחוקי המשחק "k בום!", כאשר  $n$  ו- $k$  הינם משתנים שערכם ייקבע בשתי השורות הראשונות של התוכנית. כל ערך יודפס בשורה חדשה. לדוגמה, עבור  $k=7$  ו- $n=15$ , התוכנית תדפיס בדיוק את השורות הבאות:

```
1
2
3
4
5
6
boom-boom!
8
9
10
11
12
13
boom!
15
```

### עזרה:

דרך פשוטה לבדוק האם ספרה מופיעה במספר היא המרת המספר למחרוזת, ושימוש בפקודה:

if x in y:

כאשר x ו-y הן שתי מחרוזות, והביטוי  $x \text{ in } y$  שווה True אם x מוכלת ב-y, False אחרת.

### הנחיות הגשה:

- התוכנית תוגש כאמור במקום המתאים בקובץ ה-py אותו אתם מגישים.
- התוכנית תדפיס אך ורק את הערכים הדרושים בדיוק לפי הפורמט המצויין בדרישות השאלה, ללא שום תוספות או הודעות (אפילו לא סימני פיסוק או רווחים מיותרים). על מנת לקבל את מלוא הניקוד עליכם להקפיד על מילוי הנחיה זו.

- הגישו את התוכנית שלכם עבור  $n=100$  ו- $k=3$ . כלומר שתי השורות הראשונות בתוכנית שלכם יהיו:

```
k=3
```

```
n=100
```

ולאחר מכן יתר התוכנית. כמובן, התוכנית צריכה לעבוד נכון עבור כל  $n$  טבעי וכל  $1 \leq k \leq 9$  טבעי.

- בעת בדיקת התרגילים הבודק עשוי לשנות את ערכם של  $n$  ו- $k$  בקובץ שלכם.

## שאלה 6

בשאלה זו נרצה לחשב מהו האורך של רצף מקסימלי של ספרות אי-זוגיות במספר  $7**44444$ . כמו כן נרצה לדעת באיזה אינדקס מתחיל רצף זה ומהו הרצף עצמו (אם יש כמה כאלה יודפס הראשון משמאל).  
למשל עבור המספר 233002475 האורך המקסימלי הוא 2, וישנם שני רצפים שמתאימים לאורך זה: הרצף 33 שמתחיל באינדקס 1 והרצף 75 שמתחיל באינדקס 7. במקרה זה יבחר הראשון.  
במקרה שהמספר אינו מכיל ספרות אי-זוגיות האורך המקסימלי הינו 0, האינדקס יהיה 1- והרצף יהיה None (None הוא ערך מיוחד בפייתון).  
כדי לענות על השאלה כיתבו קוד פייתון מתאים בקובץ השלד skeleton1.py.

הנחיות: המספר השלם האי-שלילי ייקלט מהמשתמש כ**מספר מפורש** (למשל 10) או כ**ביטוי אריתמטי** (למשל  $10**2$ ) באמצעות פקודת input שמופיעה כבר בקובץ השלד (רמז: שאלה 1). את יתר הקוד הוסיפו לאחר מכן.  
בסיום יודפסו התוצאות (פקודות ההדפסה כבר מופיעות בקובץ השלד ואין לשנות אותן). מותר להניח שהקלט תקין, ואין צורך לבדוק זאת, או לטפל במקרים אחרים.  
דוגמאות הרצה:

```
Please enter a positive integer: 233002475
The maximal length is 2
Sequence starts at 1
Sequence is 33
```

```
Please enter a positive integer: 2**10
The maximal length is 1
Sequence starts at 0
Sequence is 1
```

המלצה: התחילו בחישוב האורך המקסימלי בלבד, ורק אח"כ הוסיפו מה שצריך לחישוב האינדקס והרצף.  
הנחיות הגשה: את התשובות עבור המספר  $7**44444$  צרפו לקובץ ה-pdf. הקוד שכתבתם יופיע בקובץ ה-py, ויבדק על קלט שלם חיובי אחר כלשהו.

## סוף.