



## מבוא למחשב – שפת פייתון (234128)

מועד ב' סמסטר חורף תש"ף

11 במרץ 2020

**משך המבחן:** 180 דקות (שלוש שעות).  
**חומר עזר:** אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני.

### הנחיות והוראות:

- בדקו שיש 20 עמודים (4 שאלות) במבחן, כולל עמוד זה.
  - החל מעמוד 17 העמודים ריקים ומיועדים לטיטה.
  - כתבו את התשובות על טופס המבחן בלבד, במקומות המיועדים לכך. שימו לב שהמקום המיועד לתשובה אינו מעיד בהכרח על אורך התשובה הנכונה. אך ניתן לכתוב כל תשובה במעט שורות קוד.
  - יש לכתוב באופן ברור, נקי ומסודר. **אסור להשתמש בעפרון.** את עמוד השער גם חובה למלא בעט. אין לכתוב באדום בטופס הבחינה.
  - בכל השאלות, הנכם רשאים להגדיר (לממש) פונקציות עזר כרצונכם, תחת מגבלות השאלה. אלא אם צוין אחרת במפורש בשאלה.
  - אין להשתמש במשתנים גלובליים ו/או סטטיים.
  - מותר להשתמש בפונקציות שמופיעות בטבלה בעמוד הבא ובפונקציות קלט/פלט `print()`, `input()`.
  - אין להשתמש ללא מימוש בפונקציות מובנות של פייתון (למעט הפונקציות בטבלה) או בפונקציות שמומשו בכיתה, אלא אם צוין אחרת במפורש בשאלה.
  - בכל שאלה ניתן להשתמש בפונקציות המוגדרות בסעיפים קודמים, גם אם לא פותרתם סעיפים אלו, אלא אם נדרש אחרת.
  - יש לעמוד בעקרונות תכנות שנלמדו בקורס. שכפול קוד הוא סיבה להורדת ציון.
  - אין צורך לבדוק תקינות קלט אלא אם צוין אחרת במפורש בשאלה.
  - סדר כתיבת הפונקציות איננו חשוב.
- נוהל "**לא יודע**": אם תכתבו בצורה ברורה "**לא יודע/ת**" על שאלה (או סעיף) שבה אתם נדרשים לקודד, תקבלו 17% מהניקוד. דבר זה מומלץ אם אתם יודעים שאתם לא יודעים את התשובה.

צוות הקורס 234128

**מרצים:** פרופ' אהוד ריבלין, ד"ר ילנה נופברי,  
יעל ארז.

**מתרגלים:** גארי מטייב, רגב כהן, אהוד ברטה,  
רוג'ה חאיק, מוחמד נאסר, איימן אבו יונס, יונתן אלול,  
יותם אמיתי, רג'אאי חטיב.

# בהצלחה!



**רשימת פונקציות ופעולות:**

סיבוכיות זמן	פעולה
$O(1)$	<code>my_list.append(x) / my_list += [x]</code>
$O(k)$ , $k=\text{len}(\text{list2})$	<code>my_list.extend(list2) / my_list += list2</code>
$O(m)$ , $m=\text{len}(\text{new\_list})$	<code>new_list=my_list[start: stop: step]</code>
$O(1)$	<code>len(my_list)</code>
$O(1)$	<code>my_list[i]</code>
$O(n)$	<code>sum(my_list)</code>
$O(n)$	<code>min(my_list) / max(my_list)</code>
$O(n)$	<code>x in my_list</code>
$O(m)$ , $m$ is the length of range	<code>range(start, stop, step)</code>
$O(n)$	<code>enumerate(my_list)</code>
$O(1)$	<code>int(x) / round(x) / bool(x)</code>
$O(1)$	<code>abs(x)</code>
$O(1)$	<code>ord(c) / chr(x)</code>
$O(n \log(n))$	<code>my_list.sort()</code>
$O(n)$	<code>my_list.reverse()</code>



שאלה 1 (20 נק')

סעיף א' (7 נק')

מה יודפס כאשר תרוץ התוכנית הבאה?

```
def print_something(m):  
    for r in m:  
        for i in r:  
            print(i)  
            break  
    for r in m:  
        print(r)  
        break
```

```
M = [[1,2,3,4], [5, 6, 7, 8]]  
print_something(M)
```

אין צורך בהסבר.

1

5

[1,2,3,4]

סעיף ב' (6 נק')

מה יודפס כאשר תרוץ התוכנית הבאה?

```
def fun(string, k, s):  
    lst = [c.lower() for c in string]  
  
    for i, c in enumerate(lst):  
        if ord('a') <= ord(c) <= ord('z'):  
            tmp = ord(c)+s  
            lst[i] = chr(tmp)  
        else:  
            lst[i] = ""  
  
    lst = lst[k:] + lst[:k]  
    lst[0] = lst[0].upper()  
  
    return "".join(lst)
```

```
print(fun("aou!", 2, 4))
```

אין צורך בהסבר.

Yes





- 4 -

מה יודפס כאשר תרוץ התוכנית הבאה?

## אין צורך בהסבר.

- 5 -





## שאלה 2 (25 נק')

כל פעולת משיכת מזומנים בבנק כרוכה בתשלום עמלה, כאשר כל סכום משיכה הוא מספר שלם חיובי וכל עמלה היא מספר שלילי לא שלם. הבנק שומר ברשימה את כל הפעולות, זוג מספרים אחרי זוג מספרים, ולכן הרשימה צריכה להיות באורך זוגי.

דוגמה לרשימה תקינה של הבנק (נשים לב כי עבור כל פעולת משיכה ישנה עמלה):

`lst = [200, -3.43, 100, -2.12, 500, -5.76, 400, -3.43]`

עובד בבנק גילה כי הרשימה הינה באורך אי-זוגי, וכך הסיק כי נפלה טעות ברשימה: לא נלקחה עמלה עבור אחת הפעולות.

עליכם לממש את הפונקציה

```
def find_error(lst)
```

אשר מקבלת רשימה `lst` באורך אי-זוגי הכוללת משיכות ועמלות עם טעות אחת, ומחזירה את האינדקס של הטעות (ר' דוגמאות).

דוגמה 1: כאן הטעות הינה באינדקס 6 עבור הרשימה

`lst = [200, -3.43, 100, -2.12, 500, -5.76, 400, 200, -3.43]`

דוגמה 2: הפונקציה מחזירה 0 עבור הרשימה

`lst = [200, 600, -9.44, 100, -2.12, 500, -5.76, 400, -3.43]`

דרישות:

1. על הפונקציה לעבוד בסיבוכיות  $O(\log(n))$  כאשר  $n$  אורך הרשימה `lst`.

2. פתרון פחות יעיל **יקבל 5 נקודות** (כמו "לא יודע").

3. אורך הרשימה הינו אי-זוגי, ברשימה ישנה טעות אחת בדיוק.

```
def find_error(lst):
```

```
    low = 0
    high = len(lst) - 1
    while low <= high:
        mid = (low + high) // 2
        if mid % 2 == 1: #make sure we check only even indices
            mid -= 1
        if lst[mid] > 0 and lst[mid + 1] > 0:
            return mid

        if lst[mid] < 0:
            high = mid - 2
        else:
            low = mid + 2
    return -1
```



- 8 -





### שאלה 3 (30 נק')

#### סעיף א' (10 נק')

עליכם לממש את הפונקציה

```
def merge(lst1, lst2)
```

אשר מקבלת שתי רשימות של מספרים שלמים, שתיהן ממוינות בסדר עולה.  
הפונקציה מחזירה רשימה חדשה הכוללת את כל האיברים של שתי הרשימות בסדר עולה.

#### דרישות:

- על הפונקציה לעבוד בסיבוכיות זמן לינארית.
- פתרון פחות יעיל יקבל 2 נקודות (כמו "לא יודע").
- אסור להשתמש בפונקציות min/max.

```
def merge(lst1, lst2):
```

```
merged_lst = []
i,j = 0,0
while (i<len(lst1)) and (j<len(lst2)):
    if (lst1[i]<=lst2[j]):
        merged_lst.append(lst1[i])
        i+=1
    else:
        merged_lst.append(lst2[j])
        j+=1
while i<len(lst1):
    merged_lst.append(lst1[i])
    i+=1
while j<len(lst2):
    merged_lst.append(lst2[j])
    j+=1
return merged_lst
```

---

---

---

---

---

---

---

---

---

---



- 10 -



### סעיף ב' (10 נק')

עליכם לממש את הפונקציה

```
def abs_to_sorted(lst)
```

אשר מקבלת רשימה של מספרים שלמים  $lst$  שממוינת לפי ערכים מוחלטים בסדר עולה: כלומר,

$$|lst[i]| \leq |lst[i+1]|$$

לכל אינדקס חוקי. הפונקציה מחזירה את הגרסה הממוינת של  $lst$ .

**דרישת סיבוכיות:** על הפונקציה לעבוד בסיבוכיות זמן  $O(n)$  כאשר  $n$  הוא אורך של  $lst$ .

דוגמה:

עבור הרשימה  $lst = [1, -2, -3, 3, 4, 5, 6, -9, 15]$  שממוינת לפי ערכים מוחלטים

הפונקציה מחזירה את הרשימה הנ"ל:  $[-9, -3, -2, 1, 3, 4, 5, 6, 15]$

הערות:

- פתרון פחות יעיל יקבל 2 נקודות (כמו "לא יודע").
- מותר להשתמש בפונקציה מסעיף א' אפילו אם לא פתרתם אותו.

```
def abs_to_sorted(lst):
```

```
def abs_to_sorted(lst):
    pos_lst, neg_lst = [], []
    for val in lst:
        if val >= 0:
            pos_lst.append(val)
        else:
            neg_lst.append(val)
    neg_lst.reverse()
    return neg_lst + pos_lst
return neg_lst[::-1] + pos_lst
```

OR

```
def abs_to_sorted(lst):
    lst_positives = []
    lst_negatives = []
    for e in lst:
        if e > 0:
            lst_positives.append(e)
        else:
            lst_negatives.append(e)

    return merge(lst_positives, lst_negatives[::-1])
```



- 12 -



### סעיף ג' (10 נק')

הגדרה: רשימה  $lst$  מוגדרת כ-"חצי ממוינת לפי ערכים מוחלטים" אם היא מקיימת את הכללים הבאים:

1. לכל אינדקס  $i$  זוגי מתקיים:  $|lst[i]| \leq 2 * |lst[i+1]|$
2. לכל אינדקס  $i$  אי-זוגי מתקיים:  $|lst[i]| \leq 1/2 * |lst[i+1]|$

עליכם לממש את הפונקציה

```
def half_abs_to_sorted(lst)
```

אשר מקבלת רשימה  $lst$  "חצי ממוינת לפי ערכים מוחלטים" ומחזירה את הגרסה הממוינת שלה.

### דרישת סיבוכיות:

- על הפונקציה לעבוד בסיבוכיות זמן  $O(n)$  כאשר  $n$  הוא אורך של  $lst$ .
- פתרון פחות יעיל יקבל ניקוד חלקי.

### הערות:

- מותר להשתמש בפונקציה מסעיף א' אפילו אם לא פתרתם אותו.
- **חובה להשתמש בפונקציה מסעיף ב' אפילו אם לא פתרתם אותו (נוסף בזמן הבחינה).**

דוגמה: עבור הרשימה  $lst = [-1024, 600, -1500, -800, 1700, -1000, 3000]$  שהיא "חצי ממוינת לפי ערכים מוחלטים" הפונקציה מחזירה את הרשימה:

$[-1500, -1024, -1000, -800, 600, 1700, 3000]$

```
def half_abs_to_sorted(lst):
```

```
    first_abs_sorted = lst[::2]
    second_abs_sorted = lst[1::2]

    first_sorted = abs_to_sorted(first_abs_sorted)
    second_sorted = abs_to_sorted(second_abs_sorted)

    return merge(first_sorted, second_sorted)
```



- 14 -



#### שאלה 4 (25 נק')

הגדרה:

- הרשימה lst של מספרים שלמים נקראת "**פירמידה מיוחדת**" אם היא מקיימת את שני התנאים הבאים:
1. בחלקו ההתחלתי של lst היא מכילה מספרים אי-זוגיים בלבד, ועולה מונוטונית עד לאיבר המקסימלי,
  2. אחר כך היא מכילה מספרים זוגיים בלבד בסדר יורד.

דוגמה 1 לפירמידה מיוחדת  $lst = [1, 3, 11, 13, 15, 17, \underline{18}, 14, 6, 2]$

הסבר: עד המספר 18 (לא כולל) הרשימה מכילה מספרים אי-זוגיים בלבד וממוינת בסדר עולה, ואחר כך הרשימה מכילה מספרים זוגיים בלבד וממוינת בסדר יורד.

שימו לב, הערך המקסימלי ברשימה לא חייב להיות זוגי/אי-זוגי, ולא בהכרח נמצא באמצע הרשימה.

דוגמה 2 לפירמידה מיוחדת  $lst = [1, 7, \underline{101}, 14, 12, 10, 8, 6, 4, 2]$

עליכם לממש את הפונקציה הרקורסיבית

```
def make_pyramid(lst)
```

אשר מקבלת רשימה ממוינת lst ומחזירה רשימה חדשה שהיא "פירמידה מיוחדת".

למשל, עבור הרשימה  $lst = [1, 2, 3, 6, 11, 13, 14, 15, 17, 18]$  הפונקציה מחזירה את הרשימה מדוגמה 1.

הערות:

- הפונקציה חייבת להיות רקורסיבית.
- **אסור** להשתמש בלולאות כולל list comprehension. פתרון הכולל לולאות **לא מתקבל**.
- **אסור** קריאות לפונקציות פייתון למעט len(). **אסור** פונקציות עזר.

```
def make_pyramid(lst):
```

```
    if len(lst)==1):
        return lst
    prev_lst = make_pyramid(lst[1:])
    if lst[0]%2==0):
        return prev_lst + [lst[0]]
    return [lst[0]]+prev_lst
```

---

---

---

---

---

---

---

---

---



- 16 -





- 17 -



- 18 -



- 19 -



- 20 -