



קורס "תכנות וניתוח נתונים בשפת פייתון" (20606) – סמסטר 2024

שאלות חזרה לבחינה*

*אין לראות בשאלות המופיעות במסמך זה התחייבות מצד צוות הקורס לסגנון שאלות זהה בבחינות

שאלה 1

רשימה דו-ממדית של מספרים שלמים נקראת "רשימה מתפתלת" אם מתקיימים התנאים הבאים:

- הרשימה הינה רשימה ריבועית, כלומר מספר השורות שווה למספר העמודות
- ערך התא $[0][0]$ שווה ל-1
- ישנה סדרה עולה של ערכים שלמים עוקבים, החל מתא $[0][0]$ ועד לתא האחרון במערך (שורה ועמודה אחרונים) באופן מפותל, כלומר הערכים החל ממיקום $[0][0]$ בשורה הראשונה הם עוקבים בסדר עולה ובסיום השורה ימשיכו לעלות באופן עוקב באמצעות ירידה לשורה הבאה (בעמודה האחרונה) לכיוון השני וכך הלאה.

לפניכם דוגמה של "רשימה מתפתלת" a בגודל 5×5

	0	1	2	3	4
0	1	2	3	4	5
1	10	9	8	7	6
2	11	12	13	14	15
3	20	19	18	17	16
4	21	22	23	24	25

כתבו פונקציה בוליאנית בשם `is_serpentine` המקבלת כפרמטר רשימה דו-ממדית `mat` המלאה במספרים שלמים, ומחזירה `True` אם הרשימה היא "רשימה מתפתלת" ו-`False` אחרת.

מומלץ לכתוב פונקציות עזר.

שאלה 2

מטריצת היחידה (Identity matrix) (מסדר n) היא מטריצה (רשימה דו-ממדית) ריבועית בגודל $n \times n$ שורות ו- n עמודות) שהאלכסון הראשי שלה מורכב מאחדות וכל שאר המטריצה מאפסים (ויקיפדיה).

לדוגמה, המטריצה A שלהלן היא מטריצת יחידה מסדר 3 (האלכסון הראשי מודגש), והמטריצה B שלהלן אינה מטריצת יחידה, אולם אם האיבר $B[1][1]$ היה 1 ולא 2, והאיבר $B[1][3]$ היה 0 ולא 3, אז היא היתה מטריצת יחידה מסדר 4.

$B=$		0	1	2	3
0		1	0	0	0
1		0	2	0	3
2		0	0	1	0
3		0	0	0	1

$A=$		0	1	2
0		1	0	0
1		0	1	0
2		0	0	1

תת-מטריצה מרכזית של מטריצה ריבועית A היא תת מטריצה ריבועית שמרכזה מתלכד עם המרכז של A .

לדוגמה :

	0	1	2	3	4	5	6
0							
1							
2							
3				מרכז			
4							
5							
6							

המטריצה המסומנת במרכז, בגודל 3×3 היא תת-מטריצה מרכזית שמתלכדת עם המטריצה הגדולה בגודל 7×7 .

בשאלה זו נחשב את הגודל המקסימלי של התת-מטריצה המרכזית של A שהיא מטריצת יחידה.

לדוגמה עבור המטריצה A שלהלן :

	0	1	2	3	4	5	6
0	1	2	3	2	0	1	2
1	0	1	0	0	0	3	0
2	0	0	1	0	0	0	0
3	5	0	0	1	0	0	0
4	7	0	0	0	1	0	0
5	8	0	0	0	0	1	0
6	1	0	0	0	0	0	0

הגודל המקסימלי של התת-מטריצה המרכזית שהיא מטריצת יחידה הוא 3. (המטריצה הזו מובלטת).

אם למשל $A[1][5]$ היה שווה 0 (במקום 3) אזי הגודל המקסימלי של התת-מטריצה המרכזית שהיא מטריצת יחידה היה 5.

א. כתבו פונקציה בוליאנית בשם `is_identity` המקבלת כפרמטרים : רשימה דו-ממדית **ריבועית** בשם `mat` המלאה במספרים שלמים, משתנה שלם בשם `x` ומשתנה שלם בשם `size`. הפונקציה תחזיר `True` אם התת-רשימה בגודל `size` שהפינה השמאלית העליונה שלה היא `mat[x][x]` מטריצת יחידה. אחרת, תחזיר השיטה `False`.

ניתן להניח שערכי `x` ו-`size` לא מובילים לחרیגה מגבולות המטריצה. אין צורך לבדוק זאת. לדוגמה, הקריאה לשיטה `is_identity(A, 2, 3)` כלומר עבור המטריצה A שלפניכם החל ממיקום `A[2][2]` ובגודל 3, השיטה תחזיר `True`.

ב. כתבו פונקציה בשם `max_matrix` המקבלת כפרמטר רשימה דו-ממדית **ריבועית** `mat` המלאה במספרים שלמים. הפונקציה צריכה להחזיר את הגודל המקסימלי של התת-מטריצה המרכזית של `mat` שהיא מטריצת יחידה. אפשר להניח שהמטריצה `mat` היא **ריבועית** ושמספר השורות והעמודות של `mat` הוא אי-זוגי.

לדוגמה, אם המטריצה שלפניכם הייתה מועברת כפרמטר לשיטה, היה מוחזר הערך 3. **שימו לב** שאומנם יש במטריצה זו מטריצת יחידה גדולה יותר, המתחיל באיבר $A[2][2]$ והיא בגודל 4 (הפינה הימנית התחתונה שלה היא $A[5][5]$, אבל המרכז של המטריצה הזו אינו מתלכד עם המטריצה A, ולכן אינה תת-מטריצה מרכזית של A :

	0	1	2	3	4	5	6
0	1	2	3	2	0	1	2
1	0	1	0	0	0	3	0
2	0	0	1	0	0	0	0
3	5	0	0	1	0	0	0
4	7	0	0	0	1	0	0
5	8	0	0	0	0	1	0
6	1	0	0	0	0	0	0

חובה להשתמש בפונקציה `is_identity` שכתבתם בסעיף א.

שאלה 3

- א. כתבו פונקציה רקורסיבית בשם `exist` המקבלת מספר שלם `num` ורשימה של מספרים `lst`. הפונקציה תחזיר `True` אם ערך `num` נמצא ברשימה `lst`. אחרת, תחזיר `false`. ניתן להניח כי כל איברים הרשימה `lst` שונים זה מזה.
- אסור להשתמש באופרטור הלוגי `in` או להשתמש בפונקציות עזר או העמסת פרמטרים.
- ב. כתבו פונקציה רקורסיבית בשם `find_pair` המקבלת מספר שלם `sum` ורשימה של מספרים `lst`. הפונקציה תחזיר `True` אם קיימים שני איברים שונים ברשימה שסכומם שווה ל-`sum`. אחרת, תחזיר `False`.
- מומלץ לעשות שימוש בפונקציה `exist` שכתבתם בסעיף א. ניתן לעשות שימוש בפונקציות עזר רקורסיביות או העמסת פרמטרים. אסור להשתמש באופרטור הלוגי `in`.

שאלה 4

כתבו פונקציה רקורסיבית בשם `minus_plus` המקבלת רשימה של מספרים שלמים (חיוביים ושליילים בלבד ללא אפס). הפונקציה תחזיר `True` אם אורך הרשימה זוגי ולכל מספר ברשימה יש מספר תואם עם סימן הפוך ממנו הנמצא ברשימה. בכל מקרה אחר, תחזיר הפונקציה `False`.

דוגמאות:

- עבור הרשימה `[7, -2, 5, 2, -5, -7]` תחזיר הפונקציה `True`.
- עבור הרשימה `[5, -5, 3, 5]` תחזיר הפונקציה `False` מכיוון שברשימה נמצא הערך 3 אך לא -3.
- עבור הרשימה `[5, -5, 5]` תחזיר הפונקציה `False` מכיוון שאורך הרשימה הוא אי זוגי.

אין לעשות שימוש ברשימת עזר או לשנות את ערכי הרשימה שהתקבלה כפרמטר. מותר לעשות שימוש בפונקציות עזר רקורסיביות והעמסת פרמטרים אך אסור לעשות שימוש באופרטור `.in`.

שאלה 5

כתבו פונקציה בשם `max_mul2` המקבלת כפרמטר רשימה של מספרים שלמים, `lst`, חיוביים ושליילים בלבד (ללא אפס), וללא חזרות (כל מספר מופיע לכל היותר פעם אחת). הפונקציה תחזיר את המכפלה הגדולה ביותר האפשרית בין שני איברים (לא בהכרח רצופים) ברשימה. הניחו שאורך הרשימה המתקבלת הוא 2 לפחות (אין צורך לבדוק זאת).

דוגמאות:

- עבור הרשימה `[4, 3, 6, -5]` יוחזר הערך $24 (6 * 4)$
- עבור הרשימה `[-2, 7, 1, -4]` יוחזר הערך $8 ((-2) * (-4))$
- עבור הרשימה `[2, -4]` יוחזר הערך -8 (יש רק שני איברים ברשימה)

אין להשתמש ברקורסיה ואין לשנות את ערכי הרשימה, אפילו לא באופן זמני, לרבות מיון הרשימה. ניתן להשתמש בפונקציות עזר כרצונכם. הפתרון צריך להיות לינארי ביחס לגודל הרשימה (בדומה לסדר גודל של חיפוש לינארי).

שאלה 6

נגדיר: מחרוזת סודית היא מחרוזת תווים שעברה הצפנה בשיטה הבאה: את התו הראשון מגדילים ב-key (מספר חיובי, לא 0), את התו השני מגדילים ב- $key + 1$, וכן הלאה. כלומר, במחרוזת הסודית התו הראשון גדול ב-key מהתו המקורי, התו השני גדול ב- $key + 1$ מהתו המקורי, וכך הלאה.

דוגמאות:

- אם המחרוזת המקורית היא "d" ו-key הוא 1 אז לאחר ההצפנה המחרוזת הסודית היא "e".
- אם המחרוזת המקורית היא "aaaa" ו-key הוא 2 אז לאחר ההצפנה המחרוזת הסודית היא "cdef".
- אם המחרוזת המקורית היא "aa" ו-key הוא 4 אז לאחר ההצפנה המחרוזת הסודית היא "ef".

כתבו פונקציה רקורסיבית secret המקבלת שתי מחרוזות, $s1$, $s2$, שאינן בהכרח בעלות אורך זהה, ומספר שלם חיובי ממש, key. הפונקציה תחזיר True אם המחרוזת $s2$ היא המחרוזת הסודית המתקבלת מהמחרוזת $s1$. אחרת, תחזיר False.

דוגמאות:

- עבור המחרוזות $s1="x"$, $s2="y"$ ו-key = 1, השיטה תחזיר true
- עבור המחרוזות $s1="aaa"$, $s2="def"$ ו-key = 3, השיטה תחזיר true
- עבור המחרוזות $s1="kswt"$, $s2="luzx"$ ו-key = 1, השיטה תחזיר true
- עבור המחרוזות $s1="abc"$, $s2="bcd"$ ו-key = 1, השיטה תחזיר false
- עבור המחרוזות $s1="abc"$, $s2="cdef"$ ו-key = 2, השיטה תחזיר false
- עבור המחרוזות $s1="abcd"$, $s2="bdf"$ ו-key = 1, השיטה תחזיר false

מותר לעשות שימוש בהעמסת פרמטרים או בפונקציות עזר רקורסיביות. בפתרון השאלה מותר לעשות שימוש בפונקציות המובנות `ord`, `chr` ובפעולות חיבור או חיסור.

הפונקציה `chr` מקבלת ערך קוד האסקי (שלם חיובי) ומחזירה את ערך התו המייצג אותו. למשל, `chr(98)` תחזיר את התו 'b'.

הפונקציה `ord` מקבלת מחרוזת באורך אחד ומחזיר את ערך קוד האסקי המתאים לו. למשל `ord('b')` תחזיר את הערך 98.

שאלה 7

כתבו פונקציה בשם `bulls_and_cows` המקבלת כפרמטרים שתי רשימות של ספרות (1-9 בלבד), `number` ו-`guess`. הרשימה `number` מייצגת מספר במשחק "בול פגיעה" שיש לנחש והרשימה `guess` מייצגת את הניחוש. הפונקציה תחזיר את הניקוד שיש לניחוש, כאשר ספרה שהיא "בול" (הספרה בניחוש מופיעה במספר שיש לנחש, אך לא באותו מיקום) שווה נקודה אחת, וספרה שהיא "פגיעה" (הספרה בניחוש מופיעה במספר שיש לנחש בדיוק באותו המיקום במספר שיש לנחש) שווה לשתי נקודות.

הנחות :

- אורך שתי הרשימות זהה
- שתי הרשימות מאותחלות בספרות (1-9)
- כל הספרות בשתי הרשימות שונות זו מזו

דוגמה :

עבור הרשימות `number = [7, 5, 2, 1]`, `guess = [3, 5, 1, 4]`, יוחזר הערך 3 :

- הספרה 3 בניחוש (`guess`) כלל לא מופיעה במספר שיש לנחש (`number`).
- הספרה 1 בניחוש (`guess`) מופיע במספר שיש לנחש (`number`) אבל במיקום אחר ("בול", שווה נקודה אחת).
- הספרה 5 בניחוש (`guess`) מופיע במספר שיש לנחש (`number`) בדיוק באותו המיקום ("פגיעה" שווה שתי נקודות).
- הספרה 4 בניחוש (`guess`) כלל לא מופיעה במספר שיש לנחש (`number`).

אין לעשות שימוש ברשימת עזר או לשנות את ערכי הרשימה שהתקבלה כפרמטר.

מותר לעשות שימוש בפונקציות עזר רקורסיביות והעמסת פרמטרים אך אסור לעשות שימוש באופרטור `in`.

שאלה 8

נתונה רשימה `lst` מלאה במספרים שלמים ממויינת בסדר עולה ממש (אין צורך לבדוק זאת).

כתבו פונקציה בשם `print_pairs` המקבלת רשימה `lst` כנ"ל וכן מספר שלם חיובי `k`. הפונקציה צריכה להדפיס את כל הזוגות במערך שההפרש ביניהם הוא בדיוק `k` (אין צורך לבדוק ש-`k` חיובי).

שימו לב!

מדובר בהפרשים של הערכים שבתאים ולא בין האינדקסים.

דוגמאות :

עבור הרשימה $lst = [-7, -3, 0, 1, 3, 5, 12, 14, 17, 19, 25, 30]$

- עבור $k=2$ יודפסו הזוגות הבאים :

Pair Found: (1, 3)

Pair Found: (3, 5)

Pair Found: (12, 14)

Pair Found: (17, 19)

- עבור $k=6$ יודפסו הזוגות הבאים :

Pair Found: (-3, 3)

Pair Found: (19, 25)

- עבור $k=23$ לא יודפס כלום כי אף זוג איברים ברשימה שההפרש בין ערכיו הוא 23.

יש לפתור את הבעיה בסדר גודל לינארי ביחס לכמות האיברים ברשימה lst .

שאלה 9

נאמר ששורה של n מספרים שלמים מייצגת גבהים של נקודות ברכס, שיש בהם פסגות ועמקים.

כתבו פונקציה בשם `maximal_drop` המקבלת רשימה lst ובו המספרים המייצגים הגבהים האלו.

הפונקציה צריכה להחזיר את הנפילה המקסימלית בין שני מספרים (לא בהכרח בתאים עוקבים), כך שהמספר הגבוה נמצא לפני המספר הנמוך ברשימה.

דוגמאות :

- עבור הרשימה $[5, 21, 3, 27, 12, 24, 7, 6, 4]$ התשובה שתוחזר תהיה 24

- עבור הרשימה $[5, 21, 3, 22, 12, 7, 26, 14]$ התשובה שתוחזר תהיה 23

- עבור הרשימה $[5, 15, 22, 12, 17, 27, 3]$ התשובה שתוחזר תהיה 17

יש לפתור את הבעיה בסדר גודל לינארי ביחס לכמות האיברים ברשימה lst .

שאלה 10

בבית קפה שכונתי מעוניינים לפתח מערכת לניהול הזמנות.
לשם כך הוגדרו ארבע מחלקות: CashRegister, Order, Date, Time.

המחלקה **Time** מייצגת זמן, ולה שתי תכונות: שעה - `_hour` (בין 0 ל-23) ודקה - `_minute` (בין 0 ל-59).

במחלקה **Time** הוגדרו השיטות הבאות:

<code>def __init__(self, h=0, m=0)</code>	בנאי המקבל שני פרמטרים (שעה ודקה) של הזמן ומאתחל את ערכי התכונות (שעה ודקה) לערכי הפרמטרים, בהתאמה. יש להגדיר ערך ברירת מחדל של 0 במידה ולא סופק ערך ובמקרה שהערך שהועבר (מספר שעות ו/או מספר דקות) אינו חוקי.
---	--

בנוסף, הוגדרו שיטות `get` ו-`set` לכל אחת מהתכונות, ושיטת `__str__`. אין צורך לממש את השיטות והבנאי הנ"ל!

המחלקה **Date** מייצגת תאריך, ולה שלוש תכונות: יום - `_day` (מספר שלם בין 1 ל-31), חודש - `_month` (מספר שלם בין 1 ל-12) ושנה - `_year` (מספר שלם חיובי בין 4 ספרות).

במחלקה **Date** הוגדרו השיטות הבאות:

<code>def Date(self, d, m, y)</code>	בנאי המקבל שלושה פרמטרים (יום, חודש ושנה) של התאריך ומאתחל את ערכי התכונות (יום, חודש ושנה) לערכי הפרמטרים, בהתאמה.
--------------------------------------	---

בנוסף, הוגדרו שיטות `get` ו-`set` לכל אחת מהתכונות, ושיטת `__str__`. אין צורך לממש את השיטות והבנאי!

א. הגדירו מחדש (דריסה) את השיטה `__eq__` במחלקה **Date** המקבלת תאריך נוסף (`other`) ומחזירה `True` אם ערכי התאריך שלו זהים לערכי התאריך עליו הופעלה השיטה. אחרת, יחזר `False`.

המחלקה **Order** מייצגת הזמנה ולה ארבע תכונות: זמן - `_t` מטיפוס **Time**, תאריך - `_d` מטיפוס **Date**, מספר הזמנה - `order_id` (מקבל את ערכו מתוך משתנה מחלקה `_order_num` המאותחל תחילה לערך 1) ועלות - `_cost`.

במחלקה **Order** הוגדרו שיטות `get` ו-`set` לכל אחת מהתכונות, ושיטת `__str__`. אין צורך לממש אותן!

ב. כתבו במחלקה **Order** בנאי המקבל את תאריך ביצוע ההזמנה (יום, חודש ושנה), זמן ביצוע ההזמנה (שעה ודקה) ועלות. הבנאי יאתחל את תאריך ושעת ההזמנה בהתאם, יאתחל את מספר ההזמנה על פי ערך מופע המחלקה `_order_num` והעלות. במידה ולא סופקה עלות ההזמנה, יש לעדכן לערך 50. יש לעדכן את מופע המחלקה ב-1 לאחר בניית האובייקט. ניתן להניח כי ערכי הפרמטרים המתקבלים חוקיים.

ג. הגדירו מחדש (דריסה) את השיטה `__gt__` במחלקה `Order` המקבלת הזמנה נוספת (`other`) ומחזירה `True` אם עלות ההזמנה עליה הופעלה השיטה גדולה מעלות ההזמנה הנוספת. אחרת, יוחזר `False`.

המחלקה `CashRegister` מייצגת קופה בבית הקפה.
הייצוג נעשה באמצעות רשימת ההזמנות שבקופה, `_orders`.

ד. כתבו במחלקה `CashRegister` שיטה בשם `monthly_total_income` המקבלת חודש `month` ומחזירה את סך כל ההכנסות מהזמנות בחודש `month`. הניחו כי כלל ההזמנות ברשימה `_orders` נמצאים באותה שנה.

ה. כתבו במחלקה `CashRegister` שיטה בשם `most_expensive_order` המקבלת תאריך `date` ומחזירה את מספר ההזמנה עם העלות המקסימלית בתאריך `date`. הניחו שיש הזמנה יחידה עם עלות מקסימלית באותו תאריך.

ו. כתבו במחלקה `CashRegister` שיטה בשם `less_than` המקבלת עלות הזמנה, `cost`. השיטה תחזירה את רשימת ההזמנות שעלותן נמוכה מ-`cost`. במקרה ואין אף הזמנה העונה על הדרישות, יש להחזיר `None`.

שאלה 11

ברצוננו לפתח מערכת לניהול אנשי קשר במערכת רב-שיח (מסרונים) א-סינכרונית. לשם כך הוגדרו שלוש מחלקות: `Date`, `Person`, `ContactList`.

המחלקה `Date` מייצגת תאריך, ולה שלוש תכונות: יום – `_day` (מספר שלם בין 1 ל-31), חודש – `_month` (מספר שלם בין 1 ל-12) ושנה – `_year` (מספר שלם חיובי בין 4 ספרות).

במחלקה `Date` הוגדרו השיטות הבאות:

<code>def Date(self, d, m, y)</code>	בנאי המקבל שלושה פרמטרים (יום, חודש ושנה) של התאריך ומאתחל את ערכי התכונות (יום, חודש ושנה) לערכי הפרמטרים, בהתאמה.
<code>def __eq__(self, other)</code>	שיטה המקבלת תאריך נוסף (<code>other</code>) ומחזירה <code>True</code> אם ערכי התאריך שלו זהים לערכי התאריך עליו הופעלה השיטה. אחרת, יוחזר <code>False</code> .

בנוסף, הוגדרו שיטות `get` ו-`set` לכל אחת מהתכונות, ושיטת `__str__`. **אין צורך לממש את השיטות והבנאי!**

א. הגדירו מחדש (דריסה) את השיטה `__lt__` במחלקה `Date` המקבלת תאריך נוסף (`other`) ומחזירה `True` אם התאריך עליו הופעלה השיטה קודם מהתאריך הנוסף. אחרת, יוחזר `False`.

המחלקה **Person** מייצגת אדם ולה שלוש תכונות: שם – `_name` מטיפוס מחרוזת, מספר זהות – `_id` מטיפוס מחרוזת ותאריך לידה – `_birth` מטיפוס `Date`.

במחלקה **Person** הוגדרו שיטות `get` ו-`set` לכל אחת מהתכונות, ושיטת `toString`. **אין צורך לממש אותן!**

- ב. כתבו במחלקה `Person` בנאי המקבל שם, מספר זהות ותאריך לידה. הבנאי יאתחל את ערכי התכונות של איש הקשר בהתאם. ניתן להניח שכל הפרמטרים שהתקבלו מאותחלים כנדרש.
- ג. הגדירו מחדש (דריסה) את השיטה `__eq__` במחלקה `Person` המקבלת איש קשר נוסף `other` ובודקת האם איש הקשר שעליו מופעלת השיטה ואיש הקשר הנוסף שווים. נגדיר שני אנשי קשר כשווים אם שמם ותאריך לידתם שווים. ניתן להניח כי הפרמטר המתקבל מאותחל.

המחלקה **ContactsList** מייצגת רשימת אנשי קשר במאגר נתונים.

הייצוג נעשה באמצעות רשימה השומרת את רשימת אנשי הקשר `contacts` הקיימים במאגר.

- ד. כתבו במחלקה `ContactsList` שיטה בשם `born_in_date` המקבלת כפרמטר תאריך `d`, ומחזירה את כמות אנשי הקשר אשר נולדו בתאריך זה. ניתן להניח כי הפרמטר המתקבל מאותחל.
- ה. כתבו במחלקה `ContactsList` שיטה בשם `oldest_contact` המחזירה את שמו של איש הקשר המבוגר ביותר ברשימת אנשי הקשר. במידה ורשימת אנשי הקשר ריקה, יוחזר `None`. הניחו שקיים איש קשר יחיד העונה על התנאי.
- ו. כתבו במחלקה `ContactsList` שיטה בשם `born_in_month` המחזירה רשימה ובה כל איבר מייצג רשימה סטטית (`tuple`) עם שני איברים: האיבר הראשון מייצג את החודש והאיבר השני מייצג את כמות אנשי הקשר שנולדו באותו חודש. יש להקפיד להחזיר רשימה שבדר איבריה ממוינים בסדר עולה על פי מספר החודש. במקרה ואין אנשי קשר שנולדו באותו חודש, אין להוסיף לרשימה המוחזרת. במידה ואין כלל אנשי קשר יש להחזיר `None`.

בהצלחה!