



האוניברסיטה הפתוחה

**קורס 20606**

**תכנות וניתוח נתונים בשפת פייתון**

**בחינה לדוגמה 2\***

\*אין לראות בשאלות המופיעות או במבנה של בחינה זו התחייבות מצד צוות הקורס למבנה בחינה או סגנון שאלות זהה בבחינות של הסמסטר הנוכחי

## שאלה 1

**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  $3 \times 3$  היא תת-מטריצה מרכזית שמתלכדת עם המטריצה הגדולה בגודל  $7 \times 7$ .

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 2

כתבו פונקציה רקורסיבית בשם `minus_plus` המקבלת רשימה של מספרים שלמים (חיוביים ושליליים 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 `[-5, -7, 2, 5, -2, 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 3

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 `[14, 21, 1, 20, 12, 7, 22, 26]` התשובה שתוחזר תהיה 25
- עבור הרשימה `[27, 15, 12, 22, 3, 7, 5, 14]` התשובה שתוחזר תהיה 11

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

### שאלה 4

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 הפונקציות הבאות :

<code>def Date(self, d, m, y)</code>	פונקציית בנאי המקבלת שלושה פרמטרים (יום, חודש ושנה) של התאריך ומאתחלת את ערכי התכונות (יום, חודש ושנה) לערכי הפרמטרים, בהתאמה.
<code>def __eq__(self, other)</code>	פונקציה 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 פונקציות `get` לכל אחת מהתכונות, ופונקציית `__str__`. אין צורך לממש פונקציות אלו!

א. כתבו את הפונקציה `__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 פונקציות `get` לכל אחת מהתכונות, ופונקציית `__str__`. **אין צורך לממש אותן!**

ב. כתבו במחלקה **Person** פונקציית `__init__` בנאי המקבלת שם, מספר זהות ותאריך לידה. הבנאי יאתחל את ערכי התכונות של איש הקשר בהתאם. ניתן להניח שכל הפרמטרים שהתקבלו מאותחלים כנדרש.

ג. כתבו את הפונקציה `__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 **Friend** מייצגת חבר ובנוסף להיותה אדם, היא מאופיינת בדרגת קרבה (ערך שלם: 1 – מעגל חברים קרוב, 2 – מעגל חברים רחוק, 3 – חבר מהעבודה).

ד. כתבו במחלקה **Friend** פונקציית `__init__` בנאי המקבלת שם, מספר זהות, תאריך לידה ודרגת קרבה (ברירת מחדל 0). הבנאי יאתחל את ערכי התכונות של החבר בהתאם. ניתן להניח שכל הפרמטרים שהתקבלו מאותחלים כנדרש.

ה. כתבו את הפונקציה `__eq__` במחלקה **Friend** המקבלת חבר נוסף `other` ובודקת האם החבר שעליו מופעלת השיטה והחבר הנוסף שווים. נגדיר שני חברים כשווים אם שמם, תאריך לידתם ודרגת קרבתם שווים. ניתן להניח כי הפרמטר המתקבל מאותחל. **שימו לב!** יש להימנע מקוד חוזרני.

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

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

ו. כתבו במחלקה **ContactsList** שיטה בשם `born_in_date` המקבלת כפרמטר תאריך `d`, ומחזירה את כמות החברים מהעבודה אשר נולדו בתאריך זה. ניתן להניח כי הפרמטר המתקבל מאותחל.

ז. 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.

ח. 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`.