האוניברסיטה הפתוחה

קורס 20606
תכנות וניתוח נתונים בשפת פייתון

בחינה לדוגמה 1*

*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

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_serpertine המקבלת כפרמטר רשימה דו-ממדית mat המלאה במספרים שלמים, ומחזירה True אם הרשימה היא "**רשימה מתפתלת**" ו- False אחרת.

<u>מומלץ</u> לכתוב פונקציות עזר.

שאלה 2

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.
<u>אסור</u> 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.
<u>מומלץ</u> לעשות שימוש בפונקציה exist שכתבתם בסעיף א. ניתן לעשות שימוש בפונקציות עזר רקורסיביות או העמסת פרמטרים. <u>אסור</u> להשתמש באופרטור הלוגי in.

שאלה 3

כתבו פונקציה בשם max_mul2 המקבלת כפרמטר רשימה של מספרים שלמים, lst, חיוביים ושליליים 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 (24 = 4 * 6)
- עבור הרשימה [2-, 7, 1, 4-] יוחזר הערך 8 (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 4

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

המחלקה **Time** מייצגת זמן, ולה שתי תכונות: שעה - hour_ (בין 0 ל- 23) ודקה – minute_ (בין 0 ל- 59).

במחלקה **Time** הוגדרו השיטות הבאות:

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.	**def** __init__(self, h=0, m=0)

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:

פונקציית בנאי המקבלת שלושה פרמטרים (יום, חודש ושנה) של התאריך ומאתחל את ערכי התכונות (יום, חודש ושנה) לערכי הפרמטרים, בהתאמה.	**def** Date(self, d, m, y)

בנוסף, הוגדרו פונקציות get לכל אחת מהתכונות, ושיטת __str__. **אין צורך לממש פונקציות אלו!**

א. כתבו את הפונקציה __eq__ במחלקה Date המקבלת תאריך נוסף (other) ומחזירה True אם ערכי התאריך שלו זהים לערכי התאריך עליו הופעלה השיטה. אחרת, יוחזר 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 פונקציית בנאי המקבלת 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. כתבו את הפונקציה __gt__ במחלקה Order המקבלת הזמנה נוספת (other) ומחזירה True אם עלות ההזמנה עליה הופעלה השיטה גדולה מעלות ההזמנה הנוספת. אחרת, יוחזר False.

המחלקה **OnlineOrder** מייצגת הזמנה מקוונת ובנוסף להיותה הזמנה, היא מאופיינת באמצעות מחרוזת שם המשתמש של הלקוח (username) בו בוצעה ההזמנה באתר.

ד. כתבו במחלקה OnlineOrder פונקציית בנאי המקבלת את תאריך ביצוע ההזמנה (יום, חודש ושנה), זמן ביצוע ההזמנה (שעה ודקה), עלות ושם המשתמש של הלקוח שביצע את ההזמנה באתר. הבנאי יאתחל את תאריך ושעת ההזמנה בהתאם, יאתחל את מספר ההזמנה על פי ערך מופע המחלקה _order_num , העלות ושם המשתמש.

המחלקה CashRegister מייצגת קופה בבית הקפה.

הייצוג נעשה באמצעות רשימת ההזמנות (רגילות ומקוונות) שבקופה, _orders.

ה. 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 <u>באותה</u> שנה.

ו. כתבו במחלקה CashRegister שיטה בשם most_expensive_order המקבלת תאריך date ומחזירה את מספר ההזמנה המקוונת (מסוג OnlineOrder) עם העלות המקסימלית בתאריך

date. הניחו שיש הזמנה יחידה עם עלות מקסימלית באותו תאריך. אם אין הזמנה מתאימה, יש להחזיר None.

ז. 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.