

מבחן: קורס 10016 – מבוא למדעי המחשב

הנחיות

1. במבחן 4 שאלות.
2. מצורף בתיקיית המבחן קובץ `starter_y.py` שעליכם להוסיף לסביבת העבודה.
3. אין לשנות את שם הקובץ.
4. יש לעדכן בהערה בראש הקובץ את מספר הת.ז. שלכם
5. פתרון כל השאלות יהיה בקובץ זה. נתונה לכם החתימה של כל אחת מהפונקציות לשאלות השונות. **אין לשנות את שמות החתימות**. מותר לכם להוסיף פונקציות עזר נוספות.
6. בדיקות שאתם רוצים לבצע יש לכתוב **אך ורק** בפונקציית ה- `main` ולא בשום מקום אחר.

דגשים:

1. אם לא עדכנתם את מספר ה-ת.ז. יורדו 2 נקודות מציון המבחן הסופי
2. אסור לכתוב קוד שלא בתוך פונקציה. במידה ויהיה קוד כזה, יורדו 3 נקודות מהציון הסופי של הבחינה
3. מותר לכתוב/להוסיף פונקציות עזר שאינן נתונות בשלד המבחן.
 - היה וזו פונקציית עזר לתוכנית ראשית, בתהליך הבדיקות שלכם, יש לזמן את פונקציית העזר מה `main` בלבד
 - היה וזו פונקציית עזר לפתרון שאלה, יש לזמן את הפונקציה בקוד הפתרון של השאלה.

שאלה מס' 1 (20 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def twin_neighbours(my_list):
```

הפונקציה מקבלת רשימה של מספרים, ומחזירה את מספר הזוגות הסמוכים/צמודים זה לזה אשר ערכם זהה. דוגמאות:

- עבור הרשימה: [1,1]: יוחזר הערך 1
- עבור הרשימה: [1,2,1]: יוחזר הערך 0
- עבור הרשימה: [1,1,1,1,1]: יוחזר הערך 4
- עבור הרשימה: [1,1,42,42]: יוחזר הערך 2
- עבור הרשימה: [1]: יוחזר הערך 0

כתבו קוד **רקורסיבי** לפונקציה `twin_neighbours` בהתאם להגדרת השאלה.

הנחיות לפתרון:

- מותר לכתוב פונקציות עזר.
- פתרון שאינו רקורסיבי יפסל ויקבל ניקוד 0.
- אין להשתמש ב- slicing (כלומר שימוש ביצירת רשימה חדשה מטווח אינדקסים של הרשימה) ויש להעביר את הרשימה המקורית בכל קריאה רקורסיבית.
- אין להשתמש בלולאה מכל סוג שהוא בפתרון. פתרון הכולל לולאות – יפסל ויקבל ניקוד 0.
- סדר גודל זמן ריצה של הפתרון: $O(n)$ או טוב יותר. פתרון רקורסיבי שהינו פחות יעיל מסדר גודל של $O(n)$, יורדו 5 נקודות מציון השאלה.

שאלה מס' 2 (30 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def rotate_matrix_90_degrees_clockwise(matrix):
```

הפונקציה מקבלת מטריצה ריבועית בגודל $N \times N$

הפונקציה מסובבת את הערכים במטריצה 90 מעלות עם כיוון השעון (ימינה)

שימו לב: הפונקציה לא בונה/מייצרת מטריצה חדשה, אלא משנה את המטריצה שהתקבלה

הנחיות לפתרון:

- אסור להשתמש ברשימות עזר
- יש לבצע את המשימה במטריצה המקורית (!)
- אין להשתמש במנגנון slicing להפיכת רשימה
- אין להשתמש בפונקציות של רשימות, אלא אך ורק לפנות לאיברים באופן ישיר ולבצע עליהם פעולות

דוגמה:

המטריצה לאחר סיבוב 90 מעלות ימינה

	0	1	2	3	4	5
0	16	17	18	19	20	1
1	15	30	31	32	21	2
2	14	29	36	33	22	3
3	13	28	35	34	23	4
4	12	27	26	25	24	5
5	11	10	9	8	7	6

המטריצה המקורית:

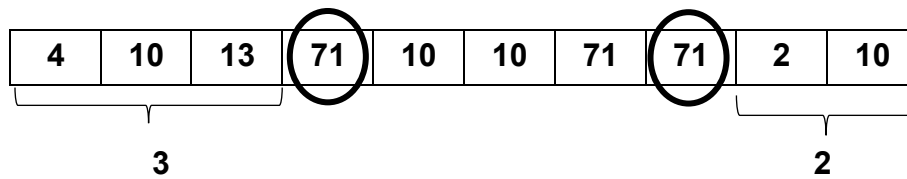
	0	1	2	3	4	5
0	1	2	3	4	5	6
1	20	21	22	23	24	7
2	19	32	33	34	25	8
3	18	31	36	35	26	9
4	17	30	29	28	27	10
5	16	15	14	13	12	11

שאלה מס' 3 (30 נקודות):

נתונה רשימה המכילה מספרים שלמים.

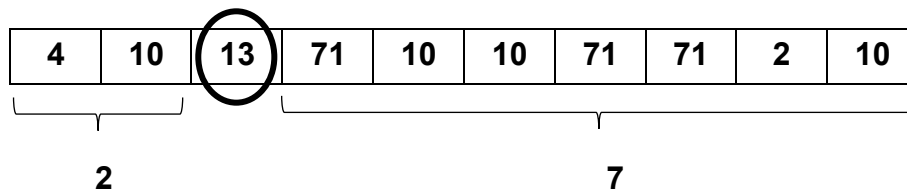
- נגדיר "**מרחק של מספר ברשימה**" כמספר הערכים שיש בינו לקצה הרשימה. לכל מספר ברשימה, יש שני מרחקים – מרחק מהקצה השמאלי שלה ומרחק מהקצה הימני שלה.
- נגדיר **סכום מרחק מינימאלי** של מספר ברשימה כסכום המרחקים הקטנים ביותר של אותו המספר מקצוות הרשימה.

לדוגמה:



- "**סכום מרחק מינימאלי**" של הערך 71: הוא 5:

- מרחק מינימאלי של 71 מהקצה השמאלי של הרשימה הוא 3
- מרחק מינימאלי של הערך 71 מהקצה הימני של הרשימה הוא 2
- ולכן הסכום הוא 5



- "**סכום מרחק מינימאלי**" של הערך 13: הוא 9:

- מרחק מינימאלי של 13 מהקצה השמאלי של הרשימה הוא 2
- מרחק מינימאלי של הערך 13 מהקצה הימני של הרשימה הוא 7
- ולכן הסכום הוא 9

סעיף 3.1 (15 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def distance_of_number_in_list (lst, num):
```

הפונקציה מקבלת רשימה של מספרים שלמים, ומספר שלם. הפונקציה מחשבת ומחזירה את "סכום מרחק מינימאלי" של הערך שהתקבל, בהתאם להגדרה לעיל. במידה והמספר לא מופיע ברשימה, יש להחזיר את הערך 1-.

סעיף 3.2 (15 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def find_minimal_distance_number(the_list):
```

הפונקציה מקבלת רשימה חד-ממדית של מספרים שלמים. הפונקציה מחזירה את המספר ברשימה שהינו בעל "סכום מרחק מינימאלי" הקטן ביותר מבין כל המספרים שברשימה.

דוגמה:

- על בסיס הרשימה בדוגמה של תחילת השאלה, **נקבל את הערך 10**.
- המרחק המינימאלי של 10 הוא 1:
 - מרחק מינימאלי של 10 מהקצה השמאלי של הרשימה הוא 1
 - מרחק מינימאלי של הערך 10 מהקצה הימני של הרשימה הוא 0 (שהרי הוא אחרון ברשימה)

הנחיות לפתרון:

- **חובה** לממש פתרון סעיף זה באמצעות זימון של הפונקציה מסעיף קודם: `distance_of_number_in_list`. מימוש סעיף זה ללא זימון מתאים של הפונקציה, יפסל ויקבל ציון 0.
- במידה ויש יותר ממספר אחד עם "סכום מרחק מינימאלי" קטן ביותר – יש להחזיר את המספר הראשון מביניהם (כלומר, המספר עם האינדקס הקטן ביותר).

שאלה מס' 4 (20 נקודות):

נתונה חתימה הפונקציה הבאה:

```
def capitalize_words(str):
```

הפונקציה מקבלת מחרוזת המכילה מילים, כאשר בין מילה למילה יש רווח אחד או יותר. כמו כן, מילה יכולה להכיל אותיות גדולות וקטנות, כולל ספרות.

הפונקציה תחזיר מחרוזת חדשה כאשר:

- כל מילה המתחילה באות – תתחיל באות גדולה
- כל שאר האותיות במילה תהינה אותיות קטנות
- ספרות ותווים מיוחדים יישארו ללא שינוי
- במידה והיו מספר רווחים בין שתי מילים, במחרוזת החדשה יהיה רווח אחד בלבד

דוגמאות:

- בהינתן המחרוזת: "welcome to my blog"
 - תוחזר המחרוזת: "Welcome To My Blog"
- בהינתן המחרוזת: "this 2Nd OPTION wITH feW SPACES and \$#COMBOs"
 - תוחזר המחרוזת: "This 2nd Option With Few Spaces And \$#combos"

כתבו קוד לפונקציה `capitalize_words` בהתאם להגדרת השאלה.