

מבחן: קורס 10016 – מבוא למדעי המחשב

הנחיות

1. במבחן 4 שאלות.
2. מצורף בתיקיית המבחן קובץ `starter_x.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 count_tens_units(num_list, num):
```

הפונקציה מקבלת רשימה של ספרות (כלומר, מספרים חד ספרתיים), ומספר דו-ספרתי. הפונקציה מחזירה את מספר הזוגות הסמוכים/צמודים זה לזה ברשימה אשר מכילים את הרצף של ספרות העשרות וספרות היחידות של המספר שהועבר.

- עבור הרשימה: [4, 2] והמספר 42: יוחזר הערך 1
- עבור הרשימה: [2, 4] והמספר 42: יוחזר הערך 0
- עבור הרשימה: [4, 1, 2, 3, 9] והמספר 42: יוחזר הערך 0
- עבור הרשימה: [4, 2, 4, 2, 4] והמספר 42: יוחזר הערך 2
- עבור הרשימה: [4] והמספר 42: יוחזר הערך 0

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

הנחיות לפתרון:

- מותר לכתוב פונקציות עזר.
- פתרון שאינו רקורסיבי יפסל ויקבל ניקוד 0.
- הפתרון צריך להתאים לכל אורך רשימה, ולכל מספר עם 2 ספרות
- ניתן להניח שערכי הפרמטרים תקינים (כלומר, אין צורך לבדוק תקינות שלהם)
- אין להשתמש בלולאה מכל סוג שהוא בפתרון. פתרון הכולל לולאות – יפסל ויקבל ניקוד 0.
- אין להמיר את הרשימה או את המספר למחרוזת בשום שלב. פתרון עם מחרוזות – 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):

	0	1	2	3	4	5
0	5	8	3	2	1	3
1	2	4	3	8	7	2
2	4	9	7	7	4	8
3	1	3	8	3	9	6

עבור רשימה דו-ממדית מלבנית (משמע כל שורותיה באורך זהה) שנתון שמכילה את הספרות 0-9 בלבד, נגדיר כי "**מספר דו-ספרתי קיים**" אם ניתן לקבל אותו ע"י רצף של 2 תאים ברשימה הדו-ממדית **לכל כיוון אפשרי**.

לדוגמה עבור הרשימה הדו-ממדית משמאל, המספר 34 קיים ברשימה באינדקסים הבאים:

- שורה 0, עמודה 2 (עם תנועה באלכסון שמאלה למטה)
- שורה 1, עמודה 2 (עם תנועה באותה שורה שמאלה)
- שורה 3, עמודה 1 (עם תנועה באלכסון שמאלה למעלה)
- שורה 3, עמודה 3 (עם תנועה באלכסון ימינה למעלה)

דוגמה נוספת עבור הרשימה הנ"ל: המספר 67 לא קיים ברשימה הדו-ממדית מפני שאין רצף של מקומות המכיל את ספרות המספר 67.

סעיף 2.1 (18 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def is_number_exist(the_list, row, col, num):
```

הפונקציה מקבלת רשימה דו-ממדית של ספרות 0-9 בלבד, אינדקס שורה, אינדקס עמודה ומספר דו-ספרתי. הפונקציה מחזירה ערך בוליאני המציין האם רצף ספרות המספר קיים במטריצה החל מאינדקס השורה והעמודה שהתקבלו.

לדוגמה, על בסיס הרשימה הדו-ממדית לעיל:

- עבור הרשימה לעיל, עבור האינדקסים (3,3) והמספר 34 הפונקציה תחזיר True.
- עבור הרשימה לעיל, עבור האינדקסים (0,5) והמספר 34 הפונקציה תחזיר False.

כתבו את קוד הפונקציה `is_number_exist` בהתאם להגדרת השאלה.

הנחיות לפתרון:

- יש לוודא שאין חריגה מגבולות הרשימה
- **ניתן להניח** שכל הערכים ברשימה הדו-ממדית הינם ספרות בלבד (כלומר בין 0-9)
- **ניתן להניח** שהמספר num שמועבר הינו מספר דו-ספרתי בלבד

סעיף 2.2 (12 נקודות):

נתונה חתימה הפונקציה הבאה:

```
def where_number_in_2dim_list(the_2dim_list, num):
```

הפונקציה מקבלת רשימה דו-ממדית של ספרות 0-9 ומספר דו-ספרתי. הפונקציה תחפש רצפים של ספרות המספר בכל הרשימה הדו-הממדית.

הפונקציה מחזירה רשימה **דו-ממדית** חדשה שכל איבר בה הוא **רשימה של 2 ערכים** המכילה את האינדקסים ברשימה הדו-ממדית, של ספרות העשרות מהמספר המבוקש, בה מתחיל רצף של הספרות. המספר הראשון בזוג הערכים הוא אינדקס השורה והמספר השני הוא אינדקס העמודה.

במידה וברשימה הדו-ממדית אין כלל את הרצף המייצג את המספר הפונקציה תוחזר רשימה ריקה.

לדוגמה, עבור הרשימה הדו-ממדית לעיל (תחילת השאלה):

- עבור הרשימה לעיל ועבור המספר 34 הפונקציה תחזיר את הרשימה הבאה:
[[0,2],[1,2],[3,1],[3,3]]

- עבור הרשימה לעיל ועבור המספר 67 הפונקציה תחזיר את הרשימה הבאה: []

כתבו את קוד הפונקציה `where_number_in_2dim_list` בהתאם להגדרת השאלה.

הנחיות לפתרון:

- **חובה** לממש פתרון סעיף זה באמצעות זימון של הפונקציה `check_number_in_list`. מימוש סעיף זה ללא זימון מתאים של הפונקציה, יפסל ויקבל ציון 0. במידה ולא מימשתם בהצלחה פונקציה זו, הניחו את קיומה.
- במקרה ובאינדקס מסוים רצף ספרות המספר מופיע בכמה כיוונים שונים, יש לציין את האינדקס פעם אחת בלבד.
- **ניתן להניח** שכל הערכים ברשימה הדו-ממדית הינם ספרות בלבד (כלומר בין 0-9)
- **ניתן להניח** שהמספר `num` שמועבר הינו מספר דו-ספרתי בלבד

שאלה מס' 3 (30 נקודות):

נגדיר את המונח "ממוצע n שכנים" ברשימה באופן הבא:

- n הוא מספר אי-זוגי
- בהינתן index של מקום ברשימה: יש לחשב ממוצע של n ערכים מהרשימה – כאשר הערך במקום הנתון הינו הערך האמצעי מבין n הערכים, ומשני הצדדים שלו יש מספר זהה של ערכים.
- במידה, והאינדקס הנתון קרוב לקצוות הרשימה, ולא ניתן לקבל מספר זהה של ערכים משני הצדדים של האינדקס האמצעי (עקב חריגה מגבולות הרשימה), "ממוצע n שכנים" יהיה הערך באינדקס הנתון.

0	1	2	3	4	5	6	7	8	9
-10	-1	8	5	1	3	8	10	12	3

נתונה הרשימה משמאל, כדוגמה. להלן מספר דוגמאות להבנת המונח "ממוצע n שכנים":

- עבור $n = 3$, ועבור האינדקס 6: הממוצע יחושב על בסיס הערכים באינדקסים 5-6-7, כלומר: $(3+8+10)/3 = 7$
- עבור $n = 5$, ועבור האינדקס 4: הממוצע יחושב על בסיס הערכים באינדקסים 2-3-4-5-6, כלומר: $(8+5+1+3+8)/5 = 5$
- עבור $n = 5$, ועבור האינדקס 7: הממוצע יחושב על בסיס הערכים באינדקסים 5-6-7-8-9, כלומר $(3+8+10+12+3)/5 = 7.2$
- עבור $n = 3$, ועבור האינדקס 0: הממוצע יהיה -10, מאחר ולא ניתן לקבל 3 ערכים, כך שמקום 0 הוא האמצעי (כי צריך מקומות 0-1-(-1), ואין ברשימה מקום (-1))
- עבור $n = 7$, ועבור האינדקס 7: הממוצע יהיה 10, מאחר ולא ניתן לקבל 7 ערכים, כך שמקום 7 הוא האמצעי (כי צריך מקומות 4-5-6-7-8-9-10, ואין ברשימה מקום 10)

סעיף 3.1 (10 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def neighbors_avg(num_list, num_neighbors, at_index):
```

הפונקציה מקבלת 3 פרמטרים: רשימה של מספרים, מספר שלם אי-זוגי המייצג מספר שכנים, אינדקס של מיקום ברשימה.

הפונקציה תחשב ותחזיר את "ממוצע n שכנים" של המקום הנתון, בהתאם להסבר ולהגדרות בתחילת השאלה.

כתבו את קוד הפונקציה `neighbors_avg` בהתאם להגדרת השאלה.

הנחיה:

ניתן להניח כי הקלט תקין והפרמטר `num_neighbors` אי-זוגי.

סעיף 3.2 (20 נקודות):

נגדיר את המונח "רשימת ממוצעי שכנים" באופן הבא:

- בהינתן רשימה מקורית של מספרים ממשיים (בכל גודל), ומספר n אי-זוגי
- "רשימת ממוצעי שכנים" הינה רשימה בגודל זהה לגודל הרשימה המקורית, וכל איבר ברשימה זו הינו "ממוצע n שכנים" של האינדקס המקביל ברשימה המקורית

	0	1	2	3	4	5	6	7	8	9	
רשימה מקורית	-10	-1	8	5	1	3	8	10	12	3	נתונה "רשימה מקורית" בגודל 10, ו-"רשימת ממוצעי שכנים" עבור n=5.
רשימת ממוצעי שכנים	-10	-1	0.6	3.2	5	5.4	6.8	7.2	12	3	

שימו לב: במקומות 0, 1, 8 ו-9 הערכים זהים למקור, ולא חושבו כממוצע, עקב הקרבה וקצוות הרשימה

נתונה חתימת הפונקציה הבאה:

```
def create_avg_neighbors(num_list, num_neighbors):
```

הפונקציה מקבלת רשימה של מספרים ממשיים, ומספר המייצג "מספר שכנים"

הפונקציה תבנה ותחשב ערכים של "רשימת ממוצעי שכנים", בהתאם להגדרות שניתנו, ותחזיר את הרשימה החדשה שבנתה.

הנחיות לפתרון:

- **חובה** לממש פתרון סעיף זה באמצעות זימון של הפונקציה `neighbors_avg`. מימוש סעיף זה ללא זימון מתאים של הפונקציה, יפסל ויקבל ציון 0. במידה ולא מימשתם בהצלחה פונקציה זו, הניחו את קיומה.

שאלה מס' 4 (20 נקודות):

נתונה חתימת הפונקציה הבאה:

```
def count_substring(text, mini_text):
```

הפונקציה מקבלת רשימה של מחרוזות, ומחרוזת נוספת – **מילה** אחת.

הפונקציה תבדוק ותחזיר כמה פעמים מופיעה **המילה** בתוך רשימת המחרוזות שהתקבלה.

שימו לב:

- מופע של מילה בתוך רשימת המחרוזות יכול להופיע גם כחלק ממחרוזת ברשימת המחרוזות.
- ייתכן שהמילה מופיעה מספר פעמים בתוך מחרוזת מרשימת המחרוזות
- ייתכן שמילה תהיה ארוכה יותר מהמחרוזת הנבדקת

דוגמאות:

- עבור הרשימה והמילה: "bc", ["abcdbcabc"] – יוחזר הערך 3
- עבור הרשימה והמילה: "bc", ["abcdbcabc", "efg", "bc"] – יוחזר הערך 4
- עבור הרשימה והמילה: "aa", ["aaaa", "abc"] – יוחזר הערך 3
- עבור הרשימה והמילה: "bccb", ["bcc", "ccb", "abccba"] – יוחזר הערך 1

כתבו את קוד הפונקציה `count_substring` בהתאם להגדרת השאלה.

שימו לב:

זוהי שאלה במחרוזות ולכן ניקוד מקסימלי יינתן על שימוש בפונקציות המתאימות הקיימות בשפה. כתיבת קוד ללא שימוש בפונקציות קיימות יגרור ציון מקסימלי של 15 נקודות עקב "המצאת הגלגל מחדש".