

# בחינה 2024ב:

## שאלה 1 (25 נקודות)

נגדיר רשימת מספרים שלמים **עולה יורדת ממש** אם בחלקה הראשון קיימת סדרת מספרים עולה ממש (כל איבר גדול מן הקודם) ובחלקה השני קיימת סדרת מספרים יורדת ממש (כל איבר קטן מן הקודם). החלוקה יכולה לקרות בכל מקום ברשימה, כולל בתחילתה או בסופה (ואז אחד החלקים הוא באורך הרשימה והשני ריק).

דוגמאות לרשימות מסוג **עולה יורד ממש**:

lst1 = [-1, 3, 7, 8, 5]

lst2 = [2, 8]

lst3 = [6, 4, 1]

דוגמאות לרשימות שאינן מסוג **עולה יורד ממש**:

lst4 = [-1, 3, 7, 8, 5, 6]

lst5 = [2, 8, 8]

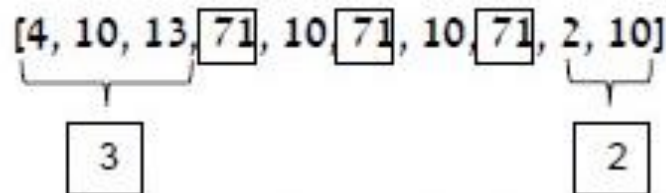
lst6 = [6, 4, 1, 2]

כתבו פונקציה בשם is\_up\_down המקבלת רשימת מספרים lst. הפונקציה תחזיר True אם lst מהווה רשימה **עולה יורדת ממש**. אחרת, תחזיר False. ניתן להניח כי הרשימה המועברת כפרמטר מאותחלת ומכילה לפחות שני איברים. אין צורך לבדוק זאת. חובה לפתור את השאלה בסדר גודל לינארי.

## שאלה 2 (25 נקודות)

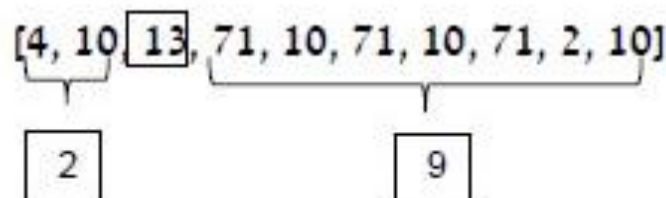
בהינתן רשימה המכילה מספרים שלמים, נגדיר "מרחק" של מספר ברשימה כסכום המרחקים הקטנים ביותר של מופע אותו המספר מקצוות הרשימה.

דוגמה 1:



ה"מרחק" של המספר 71 ברשימה הוא 5 (2 מימין ו-3 משמאל).

דוגמה 2:



ה"מרחק" של המספר 13 ברשימה הוא 9 (7 מימין ו-2 משמאל).

## סעיף א: (15 נקודות)

כתבו פונקציה בשם `dist` המקבלת כפרמטר רשימת מספרים `lst` ומספר שלם `num`. הפונקציה תחזיר ה"מרחק" של `num` ברשימה `lst`. במידה ו-`num` אינו נמצא ב-`lst`, יש להחזיר -1. ניתן להניח כי הרשימה `lst` המתקבלת כפרמטר מאותחלת כנדרש. בפתרון השאלה אסור להשתמש בשיטה `index` המוגדרת לטיפוס `list`.

## סעיף ב: (10 נקודות)

כתבו פונקציה בשם `min_dist` המקבלת כפרמטר רשימת מספרים `lst`.

הפונקציה תחזיר את המספר עם ה"מרחק" הקטן ביותר.

לדוגמה: עבור הרשימה `[4, 10, 13, 71, 10, 71, 10, 71, 2, 10]` יוחזר הערך 10 כי ה"מרחק" שלו

הוא 1, הקטן ביותר מבין כל ה"מרחקים" הקיימים.

ניתן להניח כי הרשימה `lst` המתקבלת כפרמטר מאותחלת כנדרש.

מומלץ לעשות שימוש בפונקציה `dist`, גם אם לא כתבתם אותה.

### שאלה 3 (25 נקודות)

#### סעיף א: (13 נקודות)

כתבו פונקציה רקורסיבית בשם `min_str` המקבלת רשימת מחרוזות `lst`. הפונקציה תחזיר את המחרוזת שאורכה מינימלי. ניתן להניח כי הרשימה `lst` המתקבלת אינה ריקה.

אסור לעשות שימוש בפונקציות עזר, לרבות פונקציות פנימיות.

#### סעיף ב: (12 נקודות)

כתבו פונקציה רקורסיבית בשם `search` המקבלת שתי מחרוזות `s1, s2`. הפונקציה תחזיר `True` אם `s2` קיימת ב- `s1`. אחרת, תחזיר `False`.

דוגמאות:

- עבור `s1 = "ewxabc"`, `s2 = "abc"` יוחזר `True`
- עבור `s1 = "a"`, `s2 = "aa"` יוחזר `False`

אסור לעשות שימוש בפונקציות עזר, לרבות פונקציות פנימיות וכמובן שאסור לעשות שימוש באופרטור `in` הקיים בשפה. יש להימנע מקריאות רקורסיביות מיותרות.

#### שאלה 4 (25 נקודות)

המחלקה Lesson מייצגת שיעור, ולה ארבע תכונות:

- `_id` – מספר הקורס, מטיפוס מחרוזת
- `_hour` – השעה שבה השיעור מתחיל (בין 8 ל-18), מטיפוס שלם
- `_minute` – הדקה שבה השיעור מתחיל (בין 0 ל-59), מטיפוס שלם
- `_duration` – משך השיעור בדקות, מטיפוס שלם

המחלקה WeeklyPlan מייצגת תכנון שבועי, ולה תכונה אחת:

- `_lessons` – רשימת שיעורים שבועית

במחלקה Lesson הוגדרו שיטות `get` ו-`set` לכל אחת מהתכונות. אין צורך לממש פונקציות אלו!

#### סעיף א: (5 נקודות)

כתבו במחלקה Lesson פונקציית `בנאי המקבלת את קוד הקורס, שעה ודקה בה השיעור מתחיל.`  
משך כל שיעור מוגדר ל-120 דקות כברירת מחדל. במידה וערך השעה או הדקה של תחילת השיעור המתקבל אינו חוקי, יש לאתחל לשעה 30:8.

#### סעיף ב: (5 נקודות)

כתבו במחלקה Lesson שיטה בשם `after` המקבלת שיעור נוסף, `other`. הפונקציה תחזיר `True` אם השיעור עליו מופעלת השיטה מסתיים מאוחר יותר ביחס לשיעור הנוסף. אחרת, תחזיר `False`. ניתן להניח ששני השיעורים מסתיימים באותה יממה. במידה והאובייקט המתקבל אינו מסוג Lesson, יש להחזיר `None`.

#### סעיף ג: (4 נקודות)

כתבו במחלקה WeeklyPlan פונקציה בשם `total_minutes` המקבלת מספר קורס, `id`. הפונקציה תחזיר את סך הדקות הנלמדות במהלך השבוע בקורס `id` שהתקבל כפרמטר.

#### סעיף ד: (5 נקודות)

כתבו במחלקה WeeklyPlan פונקציה בשם `most_later_lesson`. הפונקציה תחזיר את השיעור שמועד סיומו הוא המאוחר ביותר מכלל השיעורים באותו שבוע. ניתן להניח שקיים שיעור אחד כזה. במידה ואין כלל שיעורים באותו שבוע, יש להחזיר `None`.

#### סעיף ה: (6 נקודות)

כתבו במחלקה WeeklyPlan פונקציה בשם `popular_course`. הפונקציה תחזיר את מספר הקורס הנלמד הכי הרבה באותו שבוע (כלומר שסך הדקות בשיעורים שנלמדו בו הוא המקסימלי). יש להניח כי קיים קורס אחד כזה. במידה ולא יתקיימו שיעורים כלל, יש להחזיר מחרוזת `None`.



### שאלה 5 (25 נקודות)

רשימה דו-ממדית נקראת "קסומה" אם סכום ערכי האיברים הנמצאים במסגרת הרשימה שווים לסכום האיברים שאינם במסגרת.

לדוגמה: הרשימות  $mat1$  ו-  $mat2$  הבאות הן רשימות "קסומה":

```
mat1 = [[1, 1, 1, 1],
         [1, 3, 3, 1],
         [1, 3, 3, 1],
         [1, 1, 1, 1]]

mat2 = [[1, 2, 1, 1, 2],
         [1, 4, 4, 4, 1],
         [3, 3, 4, 3, 1],
         [2, 3, 1, 1, 2]]
```

הסבר:

עבור  $mat1$  סכום האיברים במסגרת זהה לסכום האיברים הפנימיים (ערכי הסכומים הוא 12).

עבור  $mat2$  סכום האיברים במסגרת זהה לסכום האיברים הפנימיים (ערכי הסכומים הוא 22).

כתבו פונקציה בשם `magic_list` המקבלת רשימה דו-ממדית `mat` המכילה איברים שלמים.

הפונקציה תחזיר `True` אם `mat` היא רשימה "קסומה". אחרת, הפונקציה תחזיר `False`.

ניתן להניח שמספר האיברים בכל שורה זהה.

תוכלו להשתמש בפונקציה המובנית `sum` הקיימת בשפה.