תרגיל בית 6
רקורסיה

הנחיות כלליות:

•קראו **היטב** את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.

•את התרגיל יש לפתור לבד!

•הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ ex1_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז שלכם, כל 9 הספרות כולל ספרת הביקורת.

•מועד אחרון להגשה: כמפורסם באתר.

•בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה, הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון וכי התוכנית אינה קורסת).

•היות ובדיקת התרגילים עשויה להיות אוטומטית, **יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח)**

•אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.

•**אין לשנות את שמות המשתנים שכבר מופיעים בקובץ השלד של התרגיל.**
יש לעבוד עם המשתנים שמופיעים בשלד התרגיל . על הקוד של כל שאלה לעבוד ולספק את התוצאה הדרושה עבור קלט שיוזן במשתנים שמופיעים בשלד (יש להחליף את סימני השאלה שמופיעים ליד שמות משתני הקלט בערכים לבחירתכם ולהגיש את התרגיל ללא סימני השאלה). יחד עם זאת, אתם רשאים להוסיף משתנים נוספים כראות עינכם.

• אין למחוק את ההערות שמופיעות בשלד .

רקורסיות קלות

שאלה 1:

ממשו את הפונקציה הרקורסיבית **reverse_string(s)** אשר מקבלת כקלט מחרוזת ומחזירה את המחרוזת בהיפוך סדר האותיות.

- ניתן להניח ש s מטיפוס str
- אין להשתמש בפונקציות מובנות של פייתון להיפוך מחרוזות או בסלייסינג עם צעד שלילי.
- אין להשתמש בלולאות.

דוגמאות הרצה :

```
>>> reverse_string("abc")
'cba'
>>> reverse_string("Hello!")
'!olleH'
```

שאלה 2:

ממשו את הפונקציה הרקורסיבית **max_rec(lst)** המקבלת רשימה לא ריקה lst של מספרים שלמים ומחזירה את המספר המקסימלי ברשימה.

הדרכה: אם lst רשימה באורך לפחות 2 , אזי המקסימום שלה הוא הגדול יותר מבין שני המספרים הבאים :

- המספר הראשון ברשימה – lst
- המקסימום של איברי הרשימה - lst החל מהאיבר השני ואילך.
- אין להשתמש בלולאות.

דוגמת הרצה :

```
>>> max_rec([3 ,5 , 5 ,1])
5
```

רקורסיות קצת פחות קלות

שאלה 3:

אסטרונאוט שחלליתו נפגעה ממטאוריט נדרש להשליך מסיפון החללית פריטים אישיים שסכום משקלם הוא בדיוק W קילוגרם על מנת להיכנס חזרה למסלול בטוח לכדור הארץ.

ממשו פונקציה **רקורסיבית** **can_return_to_earth(weights, W, K)** שתקבל את רשימת המשקלים של הפריטים האישיים שהאסטרונאוט לקח עמו לחלל ואת (weight) ואת המשקל הכולל שיש להשליך מהחללית ,(W) והחזירו True אם קיים אוסף פריטים ברשימה שסכום משקלם הוא בדיוק W ו-False אחרת.

שימו לב: האסטרונאוט יכול להשליך רק פריטים שמשקלם הבודד קטן מ - K קילוגרם היות ומנוף החללית נפגע ואינו מתפקד .

- ניתן להניח ש K וכל האיברים ברשימת המשקלים הם מספרים שלמים חיוביים (גדולים מ-0) וש- W אינו שלילי
- ניתן לבחור כל פריט רק פעם אחת
- אין להשתמש בלולאות או בפונקציות מיון מובנות של פייתון

דוגמא :

```
>>> can_return_to_earth ([5,2,6,4,2],8,5)
True
```

הסבר : הפונקציה החזירה True כי קיים ברשימה אוסף של איברים הקטנים מ-5 שסכומם הוא 8 .

```
>>> can_return_to_earth ([5,2,6,4,2],8,3)
False
```

הסבר : הפונקציה החזירה False כי לא קיים ברשימה אוסף של איברים הקטנים מ-3 שסכומם הוא 8.

שאלה 4:

נתונים 2 סוגי מטבעות – 2 ₪ ו-5 ₪, כתבו פונקציה **רקורסיבית** בשם **is_changeable(n,lst)** שמקבלת סכום יעד n וקובעת האם ניתן להגיע לסכום היעד על ידי צירוף כלשהו של שני סוגי המטבעות. הפונקציה מקבלת גם רשימה ריקה בשם lst ותשמור בה את רשימת המטבעות שסכומם הוא n במידה וקיים צירוף מתאים.

על הפונקציה להחזיר True במידה וקיים צירוף כלשהו של שני סוגי המטבעות שסכומו n בנוסף לעדכון הרשימה , lst ותחזיר False אחרת.

הבהרות:

- מכל סוג של מטבע ניתן לא לקחת בכלל או לקחת כמה שרוצים ללא מגבלה.
- אין חשיבות לסדר המטבעות ברשימה, ואם יש כמה קומבינציות ניתן להחזיר כל אחת מהן.
- אין להשתמש בלולאות.

דוגמאות הרצה :

```
>>> lst = []
>>> is_changeable(10, lst)
True
>>> print(lst)
[2, 2, 2, 2, 2]
>>> lst2 = []
>>> is_changeable(8.5, lst2)
False
>>> print(lst2)
[]
```

- שימו לב כי עבור דוגמת ההרצה הראשונה lst ,יכול להיות [5,5], פלט זה תקין גם כן.
- רמז : התחילו במימוש גרסא פשוטה יותר שרק מחזירה האם אפשר להגיע לסכום, מבלי לעדכן את רשימת המטבעות שנדרשים לשם כך.

שאלה 5:

אתם מטפסים במעלה גרם מדרגות בן (n מספר שלם וחיובי) מדרגות. בכל צעד של טיפוס אתם יכולים לבחור לעלות מדרגה אחת בלבד או שתי מדרגות בבת אחת. בכמה דרכים שונות ניתן לטפס לקצה גרם המדרגות על ידי צירופים שונים של מדרגה אחת או שתיים ?

דוגמאות :

- יש אפשרות אחת לטפס גרם של מדרגה אחת (צעד בודד של מדרגה אחת)
- יש 2 אפשרויות לטפס גרם של 2 מדרגות (שני צעדים של מדרגה, או צעד בודד של 2 מדרגות)
- יש 3 אפשרויות לטפס גרם של 3 מדרגות (3 צעדים של 1, צעד של 1 ואז צעד של 2 או צעד של 2 ואז צעד של 1. שלוש האפשרויות ניתנות לייצוג תמציתי כ- 111,12,21)
- יש 8 אפשרויות לטפס גרם של 5 מדרגות (11111,1112,1121,1211,122,2111,212,221)
- אין להשתמש בלולאות

סעיף א'

ממשו את הפונקציה **הרקורסיבית climb_combinations(n)** אשר מקבלת מספר שלם וחיובי (גדול מאפס) n ומחזירה את מספר האפשרויות השונות להגיע לקצה גרם המדרגות.

הערה: המימוש שתכתבו לפונקציה זו ייבדק עבור ערכי n הקטנים מ-40 (בסעיף הבא תעשו שימוש בממואיזציה כדי להתגבר על מגבלה זו.)

דוגמאות הרצה :

```
>>> climb_combinations(5)
8
>>> climb_combinations(10)
89
```

סעיף ב'

ממשו את הפונקציה **הרקורסיבית climb_combinations_memo(n, memo=None)** אשר בדומה לסעיף א' מקבלת את n ומחזירה את מספר האפשרויות לעלות את גרם המדרגות, אך הפעם השתמשו **בממואיזציה** על מנת לשפר את זמן הריצה של הרקורסיה .

שימו לב שהפלטים של גרסה זו של הפונקציה אמורים להיות זהים לפלטים של הפונקציה מסעיף א' עבור אותם ערכי.n

הערה: המימוש שתכתבו לפונקציה זו ייבדק גם עבור ערכי n הגדולים מ-40, וזמן הריצה עבור n<800 אמור להיות נמוך מ-3 שניות .

דוגמאות הרצה :

```
>>> climb_combinations_memo(10)
89
>>> climb_combinations_memo(100)
573147844013817084101
```

בהצלחה !