

תרגיל 5: רקורסיה מתקדמת

תאריך פרסום:	28/11
תאריך הגשה:	11/12 בשעה 23:59
מתרגל אחראי:	נוי סלומון
משקל תרגיל:	3 נקודות
מטרות העבודה:	רקורסיות מרובות קריאות, רקורסיה עם ממואיזציה.

הנחיות ספציפיות לעבודה:

עבודה זו היא עבודה שניה מתוך זוג עבודות ברקורסיה. כפי שלמדנו בכיתה, תכננו היטב כיצד 'להקטין' כל בעיה, מה תנאי העצירה, ואיך להרכיב פתרון לבעיה מפתרונות (רקורסיביים) לתתי בעיות. בשאלות שבהן נדרשת ממואיזציה - חישבו היטב מה המפתח המתאים לתיאור כל מצב בבעיה.

אלא אם נאמר אחרת:

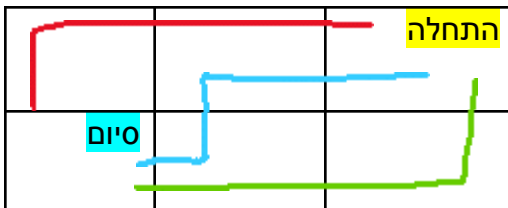
- אין להשתמש בלולאות (אתם מתרגלים רקורסיה).
- ניתן לממש פונקציות מעטפת בכל מקום בו לא נכתב בפירוש שאסור.
- בכל מקום בו אתם מגדירים פונקציית מעטפת הקפידו ששמה יהיה תואם לשם הפונקציה שנתבקשתם לממש בסעיף המתאים.
- ניתן לממש פונקציות עזר (ללא לולאות) אלא אם נאמר בפירוש אחרת. הקפידו על הגדרת שמות משמעותיים לפונקציות העזר.
- אין להשתמש בפרמטרי ברירת מחדל בהגדרה של פונקציה.
- שימו לב כי גם פקודות על אוספים כגון `sum`, `in`, `max`, `min` או `sum` מהוות לולאות ולא ניתן להשתמש בהן בתרגיל זה.

שאלה 1

סעיף א'

בהנתן מלבן שגודלו n שורות על m עמודות, נגדיר את **כמות המסלולים החוצים** להיות מספר המסלולים השונים הקצרים ביותר, היוצאים מפניה אחת של המלבן לפינתו הנגדית. מסלול חוצה מורכב מרצף של מעברים, כאשר כל מעבר הוא אופקי (ימינה\שמאלה) או אנכי (למעלה\למטה) מתא במלבן לאחד התאים השכנים. אורכו של מסלול הוא מספר התאים במלבן הכלולים במסלול.

למשל, במלבן בגודל 2 על 3, יש 3 מסלולים – מסומנים באדום, כחול וירוק (ואורכם 4):



ממשו את הפונקציה $matrix_explorer(n, m)$ המקבלת שני מספרים שלמים וגדולים מ-0 המייצגים את מימדי המלבן ומחזירה את מספר המסלולים החוצים מתא $[0,0]$ לתא $[n-1,m-1]$ במלבן.

דוגמאות:

```
>> Matrix_explorer(3,2)
3
>> Matrix_explorer(12,9)
75582
```

סעיף ב'

ממשו את הפונקציה $matrix_explorer_memo(n, m)$ עם פלט וקלט זהה לסעיף א' אולם משתמשת בממואיזציה בכדי להימנע מחישובים חוזרים עבור מלבנים בעלי מימדים גדולים.

דוגמאות הרצה: הקלטים והפלטים מסעיף א', וגם –

```
>> Matrix_explorer_memo(12,21)
84672315
```

שאלה 2

בישראל ישנם ארבעה סוגי מטבעות שונים שמייצגים ערך שלם בשקלים – מטבע שקל אחד, 2, 5 ו-10 שקלים. מחיר מסוים ניתן לייצוג באמצעות מספר קומבינציות של מטבעות בכמויות שונות, למשל – ישנן 5 דרכים שונות להציג מחיר של 6 שקלים, כמתואר בעמודות הטבלה:

כמות מכל מטבע					ערך מטבע
1	0	2	4	6	שקל
0	3	2	1	0	2 שקלים
1	0	0	0	0	5 שקלים
0	0	0	0	0	10 שקלים

ממשו את הפונקציה `count_coin_sets(money_value, list_of_coins)` שמקבלת כקלט מחיר שלם חיובי (`money_value`) ורשימה המכילה את סוגי המטבעות (`list_of_coins`), כאשר כל סוג מטבע מיוצג באמצעות ערכו הכספי (לדוגמא, `[1, 2, 5, 10]`). הפונקציה תחזיר מספר שלם המייצג את כמות הדרכים השונות לייצג את הסכום הכספי באמצעות מטבעות בסכומים המופיעים ברשימה, באמצעות רקורסיה מבלי לחזור על חישובים שכבר נפתרו.

הנחייה - בכדי להימנע מספירה כפולה של אותו ייצוג מספרי (בקריאות רקורסיביות שונות) - עליכם למיין את רשימת המטבעות (חישבו - כיצד זה עוזר?). לצורך כך, **מוותר** להשתמש בפונקציית המיין המובנית (`sorted`) אך ורק בשאלה זו, ואך ורק על רשימת סוגי המטבעות.

דוגמאות הרצה:

```
>>>count_coin_sets(6, [10,5,2,1])
```

```
5
```

```
>>>count_coin_sets(12, [10,5,2,1])
```

```
15
```

שאלה 3

ליהי ונפתול, אחיה הקטן, משחקים במשחק 'ראש זנב' אותו הם המציאו יחדיו. השחקנים בוחרים מילה כלשהי, שמוגדרת להיות "מילת המשחק". כל שחקן בתורו בוחר האם לקחת את האות הראשונה או האחרונה במילת המשחק, מוסיף לקופתו את הערך הגימטרי של האות שבחר (ראו הדרכה), ומילת המשחק עוברת ליריבו ללא האות שנבחרה. המשחק מסתיים כאשר מילת המשחק היא המילה הריקה, והמנצח הוא השחקן שסך הניקוד שצבר בקופתו גדול מזה של יריבו, אחרת תוצאת המשחק היא תיקו.

כאשר ליהי ואחיה משחקים, ליהי תמיד בוחרת את מילת המשחק (כי האנגלית שלה טובה יותר) ומבצעת את המהלך הראשון (כי היא גדולה יותר). האסטרטגיה של מפתול היא חמדנית – לקחת את האות שערכה גבוה יותר מבין שתי האפשרויות. מכיוון שנפתול, מתייאש ובוכה כאשר הוא לא מנצח פעמיים ברצף, ליהי החליטה לבחור רשימה של מילים 'בטוחות', כאלה שנפתול ינצח בהן ללא תלות במהלכים שלה.

לדוגמא, המילה 'asa' היא מילה בטוחה משום שעבור כל בחירה של ליהי, נפתול יבחר את האות s וינצח במשחק.

ממשו את הפונקציה `rosh_zanav(word)`

שמקבלת משתנה word מטיפוס מחרוזת המורכבת מאותיות ב-lowercase בלבד, ומחזירה ערך בוליאני: True אם המילה בטוחה ו-False אחרת.

הדרכה - ערך גימטרי של אות

גימטריה נותנת ערך מספרי לכל אות לפי המיקום שלה באלפבית.
 $a=1, b=2, c=3$ וכו'. חשבו כיצד ניתן לחשב זאת בצורה עקבית ומהירה!

דוגמאות הרצה:

```
>>>rosh_zanav('asa')
```

```
True
```

```
>>>rosh_zanav('evyatar')
```

```
True
```

```
>>>rosh_zanav('breakfast')
```

```
False
```