

תרגילים בסיסיים בתכנות

שאלה 1:

בהינתן רשימה, יש לספור את מספר המופעים של כל אובייקט ברשימה על ידי מילון ולבסוף להדפיס את המילון. בשאלה זו אין להשתמש בפונקציה `count`.
לדוגמא:

```
input_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]
```

הדפסה שתופיע למשתמש:

```
{11: 2, 45: 3, 8: 1, 23: 2, 89: 1}
```

שאלה 2:

פונקציה המקבלת רשימת מספרים ומספר (הקטן מאורך הרשימה). הפונקציה מחזירה את n המספרים המקסימיים של הרשימה. לא ניתן להשתמש בפונקציית מיון `sort` ו-`max`, יש לבצע זאת בלולאה.

לדוגמא:

```
input_list = [2, 6, 41, 85, 0, 3, 7, 6, 10]
number_of_top_largest = 3
print(n_max_elements(input_list, number_of_top_largest))
```

נקבל:

```
[10, 41, 85]
```

עבור:

```
input_list = [2, 6, 41, 85, 0, 3, 7, 6, 10]
number_of_top_largest = 2
print(n_max_elements(input_list, number_of_top_largest))
```

נקבל:

```
[41, 85]
```

שאלת בונוס – הגנה על הקוד:

הוסף בדיקות לקלט (האובייקטים המתקבלים) של הפונקציה משאלה 3, והגן על הקוד מקריסה במהלך הפעולות אשר מתבצעות על הקלט.
פקודות שימושיות: `try`, `except`, `raise Exception`, `assert` וכדומה.

שאלת בונוס – רקורסיה – טיפוס מדרגות:

היו היה בית עם גרם מדרגות פנימי וילד שאוהב מאוד לטפס עליו. הילד יכול לבחור לטפס מדרגה-מדרגה, או לדלג שתי מדרגות או לדלג שלוש מדרגות. הילד שונא לחזור על דרכים. אם הילד בוחר כל יום לטפס בדרך אחרת, תוך כמה ימים הוא יצטרך לעבור דירה? (כי לא יישארו לו עוד דרכים חדשות לטפס).

אם לדוגמא יש בבית 3 מדרגות אז האפשרויות של הילד די מוגבלות:

הוא יכול לטפס את כל שלושת המדרגות בצעד אחד. [3]
הוא יכול לטפס את שתי המדרגות הראשונות בצעד אחד, ואז עוד צעד למדרגה האחרונה. [1, 2]
הוא יכול לטפס את המדרגה הראשונה בצעד אחד, ואז את השתיים הנותרות בצעד הבא. [2, 1]
הוא יכול לטפס צעד-צעד את כל המדרגות ולעלות את שלושת המדרגות ב-3 צעדים. [1, 1, 1]
סך הכל אחרי 4 ימים הוא יצטרך למצוא בית חדש, אלא אם כן ההורים יצליחו לשכנע אותו שדווקא לא נורא לטפס כמה פעמים באותה שיטה.
כלומר התשובה ל- `print(ways(3))` תהיה 4.