

חשיבה מחשובית ותכנות בשפת פייתון

תרגיל בית 7

הנחיות כלליות:

- קראו **היטב** את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על הוראות ההגשה המפורסמות במודל. בפרט, יש להגיש את כל השאלות יחד בקובץ `ex#_012345678.py` המצורף לתרגיל, לאחר החלפת ה# במספר התרגיל והחלפת הספרות 012345678 במספר תז שלכם, כל 9 הספרות כולל ספרת הביקורת. למשל, אם מספר תעודת הזהות שלי הינו 112233445 ואני מגיש את תרגיל מספר 1, שם הקובץ שאגיש יהיה `ex1_112233445.py`.
- 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 **פליטים מדויקים על פי הדוגמאות (כן כן, עד לרמת הרווח)**.
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- **יש לעבוד עם המשתנים שמופיעים בשלד התרגיל ואין לשנות את שמם**. על קטע הקוד של כל שאלה לעבוד ולספק את התוצאה הדרושה עבור קלט שיוזן במשתנים שמופיעים בשלד (המשתנים שלידם סימני שאלה ומחכים לקלט כפי שמופיע בדוגמאות בתרגול). יחד עם זאת, אתם רשאים להוסיף משתנים נוספים כראותם עינכם.
- מומלץ להתעדכן בפורום לגבי שאלות של סטודנטים אחרים וכמובן, במידה ועדיין משהו לא ברור, לשאול בעצמכם.

שימו לב! (תקף החל מתרגיל בית מספר 5 ואילך)

- ✓ שימו לב 1, החתימה של כל אחת מהפונקציות שעליכם לממש מופיעה כבר בשלד התרגיל ואין לשנותה. עליכם לממש את גוף הפונקציה בלבד. יש למחוק את הפקודה pass (היא נמצאת שם רק כדי שתוכלו להריץ חלקי קוד בלי לשים בהערה את המימושים החסרים) ולהחליף אותה במימוש המתאים.
- ✓ שימו לב 2, בשאלה בה נתבקשתם לממש פונקציה, אין לעשות שום דבר מעבר לכך. ניתן ואף רצוי לקרוא לפונקציה שכתבתם עם מגוון קלטים על מנת לוודא את תקינותה אך אין להשאיר את הקריאות הללו בהגשת התרגיל.
- ✓ שימו לב 3, אף פונקציה לא מדפיסה דבר אלא רק מחזירה ערך כלשהו.

שאלה 1 – ממשו את הפונקציה `range_dict(a, b)` המקבלת שני מספרים שלמים ($a < b$) ומחזירה מילון המכיל מיפויים מ x כמחרוזת ל x כאיבר בודד ברשימה, כאשר x הוא כל מספר בטווח מ a עד b כולל. להלן דוגמת הרצה:

```
>>> d1 = range_dict(-2, 1)
>>> print(d1)
{'0': [0], '-2': [-2], '1': [1], '-1': [-1]}
```

שאלה 2 – ממשו את הפונקציה `merge_dictionaries(d1, d2)` המקבלת שני מילונים ומחזירה מילון המכיל את המיפויים של שני המילונים הני"ל. ניתן להניח שאין מפתחות המופיעים בשני המילונים. להלן דוגמת הרצה:

```
>>> d3 = merge_dictionaries({'2': 3}, {'4': 5, 'a': []})
>>> print(d3)
{2: 3, '4': 5, 'a': []}
```

שאלה 3 – ממשו את הפונקציה `most_popular_digit(num)` המקבלת מספר שלם (`int`) גדול מ0 ומחזירה אובייקט מטיפוס `int` המייצג את הספרה שמופיעה בו הכי הרבה פעמים (אם יש יותר מאחת, ניתן לבחור באופן שרירותי) להלן דוגמת הרצה:

```
>>> result = most_popular_digit(234552222)
```

```
>>> print(result)
```

```
2
```

שאלה 4 – ממשו את הפונקציה `most_common_value_in_dict(d)` המקבלת מילון ומחזירה את הערך (`value`) הנפוץ ביותר במילון (אם יש יותר מאחד, ניתן לבחור באופן שרירותי). ניתן להניח ש-`d` אינו ריק ושהערכים שבו הם `immutable` (אהמ, רמז...). להלן דוגמת הרצה:

```
>>> result = most_common_value_in_dict({'1': 2, 'a': 3, 'x': 3, 4: 3, 2: 2,
1: 'y'})
```

```
>>> print(result)
```

```
3
```

שאלת בונוס (5 נק'): בידינו מילון המכיל מיפוי בין סטודנט לרשימת הקורסים שהסטודנט רשום אליהם. ברצוננו לייצר רשימה של סטודנטים לכל קורס. ממשו את הפונקציה `swap_students_courses(student2courses)` המקבלת מילון אשר המפתחות בו הם שמות של סטודנטים, ולכל סטודנט, הערך הוא רשימה של מחרוזות אשר מייצגות את הקורסים שהסטודנט רשום אליהם. הפונקציה תחזיר מילון שהמפתחות בו הינם שמות הקורסים, ולכל קורס, הערך יהיה רשימת הסטודנטים שרשומים אליו. שימו לב, אין חשיבות לסדר ברשימות בקלט ובפלט. הניחו כי שמות הקורסים והסטודנטים הינם `case-sensitive`, כלומר הקורס "Math" שונה מהקורס "math". להלן דוגמת הרצה:

```
>>> student2courses = {"Yuval": ["Math", "Computer Science",
"Statistics"], "Gal": ["Algebra", "Statistics", "Physics"], "Noam":
["Statistics", "Math", "Programming"]}
>>> course2students = swap_students_courses(student2courses)
>>> print(course2students)
{'Math': ['Yuval', 'Noam'], 'Computer Science': ['Yuval'], 'Statistics':
['Yuval', 'Gal', 'Noam'], 'Algebra': ['Gal'], 'Physics': ['Gal'],
'Programming': ['Noam']}
```