**תרגיל בית 2**

הנחיות כלליות:

* קראו בעיון את השאלות והקפידו שהתוכניות שלכם פועלות בהתאם לנדרש.
* את התרגיל יש לפתור **בזוגות**!
* אופן ביצוע התרגיל: שימו לב, בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף. וכן לענות על שאלות פתוחות. **בנוסף עליכם להגיש קובץ PDF כפי שהוסבר בתרגול.**
* הקפידו על הגשת הקובץ המתאים. בפרט, יש להגיש את כל השאלות יחד בקובץ ex2\_012345678\_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז. שלכם (של שני הסטודנטים), כל 9 הספרות כולל ספרת ביקורת .
* הערה: חלק מהשאלות נבדקות באופן אוטומטי. לכן, **עליכם לרשום את הקוד שלכם אך ורק במקומות המתאימים לכך בקובץ השלד**. הקוד המופיע בחלק מספר ## 0. של קובץ השלד הינו דוגמא המציגה היכן עליכם לרשום את הקוד שלכם – את הפתרון שלכם לשאלה מספר k עליכם לרשום במקום המתאים לכך בחלק מספר של ## k. של קובץ השלד.
* **שימו לב לממש בצורה מדויקת אחר המתבקש בכל פונקציה (במיוחד יש להמנע ממקרים בהם התבקשתם להחזיר ערך ואתם בוחרים להדפיס אותו שלא לצורך)**
* **אין** **לשנות שמות פונקציות או משתנים** שקיימים בקובץ השלד של התרגיל. **מותר לשנות ערכי משתנים** (ואף מומלץ לעשות זאת, על מנת לוודא שהקוד שלכם עובד כשורה).
* בדיקה עצמית: ודאו את נכונותן של הפונקציות אותן אתם כותב
* ים ואת עמידותן גם למקרי קצה.
* ניתן להניח כי הפונקציות מקבלות קלט תקין, אלא אם כן נכתב אחרת.
* 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.

**שאלה 0**

ממשו את הפונקציה reverse\_number המקבלת מספר שלם אי שלילי n ומחזירה מספר שספרותיו הן הספרות כמו במספר n , אך מסודרות בסדר הפוך.

>>> reverse\_number(123)

321

**שאלה 1**

ממשו פונקציה בשם sum\_list המקבלת רשימה שלמה של מספרים שלמים lst ומחזירה את הסכום שיתקבל מחיבור כל איברי הרשימה. אין להשתמש בפונציה המובנית sum .

>>> sum\_list([1,5,3,1])

10

**שאלה 2**

ממשו פוקנציה בשם find\_strings\_with\_digit המקבלת רשימה של מחרוזות string\_list וספרה d (מספר שלם שערכו בין 0 ל-9). הפונקציה תחזיר רשימה חדשה המכילה את כל המחרוזות מstring\_list שהספרה d מופיעה בהן וכל זאת מבלי לשנות את הרשימה string\_list או את מי מאיבריה.

>>> find\_strings\_with\_digit([“abc”, “ab4”, “a4b”, “444”], 4)

[‘ab4’, ‘a4b’, ‘444’]

>>> find\_strings\_with\_digit([“abc”, “ab4”, “a4b”, “444”], 0)

[]

**שאלה 3**

ממשו פונקציה בשם validate\_matrix המקבלת את הארגומנטים הבאים (לפי הסדר הבא):

* רשימה של רשימות בשם mat
* מספר שלם חיובי m
* מספר שלם חיובי n

הפונקציה תבדוק האם mat היא מטריצה בעלת m תתי רשימות (שורות) כאשר כל תת רשימה היא בעלת n איברים (עמודות) מסוג int בלבד. אם אכן יש בדיוק m שורות ו-n עמודות של int הפונקציה תחזיר True בכל מקרה אחר False.

* ניתן להניח ש-mat היא רשימה של רשימות.

>>> validate\_matrix([[1,2,3], [3,4,5]], 2, 3)

True

>>> validate\_matrix([[1,2], [3,4,5]], 2, 2)

False

>>> validate\_matrix([[1,2], [3,4]], 3, 2)

False

**שאלה 4**

ממשו פונקציה בשם transpose\_matrix המקבלת מטריצה mat ומחזירה את מטרית השחלוף שלה (transpose). כלומר, אם המטריצה המקורית,mat, היא בעלת m שורות (תתי רשימות) בעלות n איברים (עמודות) אזי המטריצה המוחזרת תהיה בעלת n שורות ו-m עמודות והערך במיקום ה-[i][j] בmat יהיה במיקום ה-[j][i]במטריצה המוחזרת.

* ניתן להניח ש-mat היא רשימה של רשימות בעלת מימדים חיוביים (לפחות שורה אחת ועמודה אחת)

>>> transpose\_matrix([[1,2,3], [4,5,6]])

[[1, 4], [2, 5], [3, 6]]

שאלה 5

**בשאלה זו עליכם לממש את הפונקציות הבאות:**

* **generate\_random\_list המקבלת מספר שלם חיובי n ומחזירה רשימה באורך n ובה ערכים אקראיים בין 0 ל-n.**
* **binary\_list\_search המקבלת רשימה lst וכן מספר שלם val. במידה והרשימה מכילה את הערך val הפונקציה תחזיר את האינדקס שלו, אחרת תחזיר את הערך -1.**

**לאחר מימוש שתי הפונקציות כתבו בקובץ נפרד החל מאיזה גודל רשימה החיפוש הבינארי היה יעיל יותר מן החיפוש הליניארי.**

**כיצד יש לבדוק את זמן הריצה של פונצקיית חיפוש על רשימה בגודל n?**

1. **יש להגריל רשימה של מספרים על ידי generate\_random\_list**
2. **יש למצע את זמן הריצה על פני 100 הרצות, כאשר כל הרצה מבצעת:**

**2.1 הגרל מספר בין 1 ל n, חפש אותו ברשימה ושמור את זמן החיפוש.**

**(שימו לב – כל ההרצות מתבצעות על אותה רשימה שהגרלתם פעם אחת, רק המספר שאותו מחפשים משתנה מהרצה להרצה).**

**השתמשו בכל מקור מידע העומד לרשותכם כדי למדוד את משך הזמן שיקח החיפוש.**

**בהצלחה!**