מבוא למדעי המחשב 67101 – סמסטר ב' 2022

תרגיל 2 - תנאים

להגשה בתאריך **23/03/2022** בשעה 22:00

# מבוא

בכל סעיף להלן מתוארת בעיה.

- את הפתרון לכל הבעיות ממשו בקובץ על פי השם הנתון בשאלה.
- את הפתרון לכל סעיף ממשו **בפונקציה נפרדת** ותנו לה שם כמופיע **בסעיף בדיוק.**
- בסעיפים הדורשים הדפסות למסך, עשו **שימוש מדויק במחרוזות המסופקות** (לא מומלץ לעשות העתק-הדבק מקובץ זה).
    - כל הוראת הדפסה למסך מתייחסת להדפסה בשורה חדשה.
- אין להדפיס הודעות מעבר לאלו המצוינות כדרישה מפורשת.
- בפרט התרגיל המוגש צריך להכיל רק את מימושי הפונקציות המבוקשות **ולא להכיל קריאות להן**.
- בכל הסעיפים **ניתן להניח כי הקלט לפונקציות הוא תקין למעט כאשר מצוין אחרת באופן מפורש** (כלומר ניתן להניח כי הפונקציה מקבלת את מספר הפרמטרים הרצוי והפרמטרים הם מטיפוס מתאים).
- הדוגמאות הניתנות עבור הפונקציות השונות בתרגיל זה לא מכסות את כל המקרים האפשריים, ונדרש מכם לחשוב על מקרים נוספים (ובפרט מקרי קצה) שעליהם רצוי שתבדקו את הפונקציות.
- אין להשתמש בפונקציה eval בפתרון תרגיל זה.

# כתיבת הפונקציות

**1. פונקציה המקבלת פרמטרים ומחזירה ערך**:

טוני סטארק בונה לעצמו חליפה משוריינת חדשה שתשמש אותו בזמן ביצוע תפקידו כגיבור העל "איירון מן". החליפה תהיה חכמה מאוד ותצויד בטכנולוגיה מתקדמת מאוד ומחשב חכם, שידע לבצע חישובים מתמטיים מסובכים. טוני מעוניין בתור התחלה שהחליפה שלו תדע לחשב חישובים מתמטיים פשוטים, ועליו לכתוב את הפונקציה הנכונה לשם כך. כתבו פונקציה המקבלת שני מספרים ואחת מארבע הפעולות המתמטיות המופיעות לעיל ומחזירה את תוצאת החישוב.

<u>הנחיות:</u>

- את הפונקציה יש לממש בקובץ ששמו calculate_mathematical_expression.py.
- שם הפונקציה הוא calculate_mathematical_expression.
- הפונקציה מקבלת שלושה פרמטרים:
  - פרמטרים ראשון ושני: מספר (שלם int, או שבר float).
  - פרמטר שלישי: אחת מהמחרוזות {'+', '-', '*', ':'}.

    '+' עבור חיבור

    '-' עבור חיסור

    '*' עבור כפל

    ':' עבור חילוק
- הפונקציה מחזירה את ערך החישוב של שני המספרים בעזרת פעולת החשבון הנתונה <u>ולא מדפיסה דבר למסך</u>.
- בפעולות החשבון בהן יש חשיבות לסדר:
  - חיסור: יש לחסר את המספר השני מהמספר הראשון.
  - חילוק: יש לחלק את המספר הראשון במספר השני. תוצאת החלוקה צריכה להיות של שבר (בפרט, התוצאה לא צריכה להיות חלוקה ללא שארית).
- לדוגמא:
  - קריאה לפונקציה באופן הבא:

```
calculate_mathematical_expression(5, 6, '+')
```

    תחזיר את הערך 11 ולא תדפיס למסך דבר.
  - קריאה לפונקציה עם הערכים 10 (ראשון), 6 (שני) ו - '-' לא תדפיס למסך דבר ותחזיר את הערך 4.
  - קריאה לפונקציה עם הערכים 5,6 ו - '*' לא תדפיס למסך דבר ותחזיר את הערך 30.
  - קריאה לפונקציה עם הערכים 5 (ראשון), 4 (שני) ו - ':' לא תדפיס למסך דבר ותחזיר את הערך 1.25.
  - קריאה לפונקציה עם הערכים 10 (ראשון), 2 (שני) ו - ':' לא תדפיס למסך דבר ותחזיר את הערך 5.0.
- במקרה ומתקבלת פקודה לא חוקית על הפונקציה להחזיר את הערך <u>None</u>. פקודות לא חוקיות הן:
  - חלוקה באפס.

- פרמטר שלישי שאינו אחת המחרוזות {'+', '-', '*', ':'}.
- ניתן להניח כי שני הפרמטרים הראשונים הם מספרים והפרמטר השלישי הוא מחרוזת.

**2. פונקציה המשתמשת בפונקציות אחרות**:

במקביל לפיתוח החליפה של טוני, ברוס באנר עורך מחקר חשוב בנושא קרני גאמא. הוא נדרש לבצע חישובים מתמטיים בשביל המחקר בעצמו, אך הוא מאמין שהוא חכם יותר מטוני, ולכן הוא מעוניין בתוכנה פשוטה יותר וחכמה יותר לשם כך. הוא מעוניין בתוכנה שתקבל תרגיל מתמטי מלא כמחרוזת ותחזיר את תשובתו. כתבו פונקציה המקבלת מחרוזת המכילה שני מספרים המופרדים בפעולת חשבון ומחזירה את הערך המחושב של הביטוי החשבוני.

<u>הנחיות:</u>

- את הפונקציה יש לממש באותו הקובץ של הסעיף הקודם (calculate_mathematical_expression.py).
- שם הפונקציה הוא calculate_from_string.
- הפונקציה מקבלת פרמטר יחיד - מחרוזת המכילה: מספר, התו רווח, אחד מסימני הפעולות המתמטיות {'+', '-', '*', ':'}, התו רווח, מספר נוסף.
  - המספרים עשויים להינתן בכתיב עם נקודה עשרונית (לדוג': 4.2, 7.0-) או ללא נקודה עשרונית (לדוג': 15, 6).
- הפונקציה מחזירה את ערך החישוב של הביטוי המתמטי על פי סדר המספרים הנתון במחרוזת ולא מדפיסה דבר למסך.
- בפעולות החשבון בהן יש חשיבות לסדר: יש לחשב משמאל לימין (כרגיל בביטויים מתמטיים).
- לדוגמא:
  - קריאה לפונקציה עם המחרוזת '7 – 2' לא תדפיס למסך דבר ותחזיר את הערך 5.0-.
  - קריאה לפונקציה עם המחרוזת '2 – 7' לא תדפיס למסך דבר ותחזיר את הערך 5.0.
  - קריאה לפונקציה עם המחרוזת '10 : 4' לא תדפיס למסך דבר ותחזיר את הערך 0.4.
  - קריאה לפונקציה עם המחרוזת '4 : 10' לא תדפיס למסך דבר ותחזיר את הערך 2.5.
- בפתרון השאלה יש להשתמש בפונקציה מסעיף 1.
- בפתרון השאלה אין להשתמש בפונקציה eval.
- לצורך פתרון הסעיף, אפשר להשתמש בפונקציה split על מנת לבצע את החלוקה לתת מחרוזות רלוונטיות.
- ניתן להניח כי המחרוזת המתקבלת היא בפורמט של: מספר, התו רווח, מחרוזת ללא רווחים, התו רווח, מספר נוסף. בדומה לסעיף הקודם - אם המחרוזת מייצגת חישוב לא חוקי, על הפונקציה להחזיר את הערך None.

**3. פונקציה המקבלת פרמטרים ומחזירה שני ערכים**:

בקרב הנוקמים התפתח ויכוח מי הנוקם החזק ביותר. ת'ור מאמין כי הוא הנוקם החזק ביותר, אך טוני מאמין כי זהו למעשה ברוס. סטיב רוג'רס מציע פתרון פשוט, בו כל אחד יחליט מבין שלושה נוקמים כלשהם, מי מביניהם הוא הנוקם החזק ביותר ומיהו החלש ביותר. כתבו פונקציה המקבלת שלושה ערכים מספריים ומחזירה שני ערכים – הראשון: הגדול ביותר, והשני: הקטן ביותר.

<u>הנחיות</u>:

- את הפונקציה יש לממש בקובץ ששמו largest_and_smallest.py.
- שם פונקציה הוא largest_and_smallest.
- הפונקציה מקבלת שלושה פרמטרים מספריים, אשר יכולים להיות חלקם או כולם שווים בערכם.
- הפונקציה מחזירה שני ערכים - הערך הראשון הוא הגדול ביותר מבין השלושה והשני הוא הקטן ביותר מבין השלושה.
- לדוגמה:
  - קריאה לפונקציה עם הערכים 1, 5, 10 צריכה להחזירה שני ערכים, הראשון 10 והשני 1.
  - קריאה לפונקציה עם הערכים 8, 2, 7.3 צריכה להחזירה שני ערכים, הראשון 8 והשני 2.
- קריאה לדוגמה לפונקציה:

```
max_val, min_val = largest_and_smallest(5, 1, 10)
```

- בפתרון השאלה אין להשתמש בפונקציות max, min ו sort.
- ניתן להניח כי הקלט לפונקציות הוא תקין.

**4. פתרון משוואה ריבועית**:

בינתיים פיטר פארקר הוא עדיין תלמיד בתיכון ולומד בדיוק איך לפתור משוואות ריבועיות. משוואה ריבועית היא משוואה במשתנה אחד מהצורה: $ax^2 + bx + c = 0$ כאשר a, b ו - c הם מספרים ממשיים. פתרון משוואה ריבועית הוא הצבה של ערכי - x אשר מקיימים את המשוואה. למשוואה ריבועית יכולים להיות בין 0 ל - 2 פתרונות ממשיים. כתבו פונקציה המקבלת את שלושת מקדמי המשוואה ומחזירה את פתרונותיה.

<u>הנחיות</u>:

- את הפונקציה יש לממש בקובץ ששמו quadratic_equation.py.
- שם הפונקציה הוא quadratic_equation.
- הפונקציה מקבלת שלושה פרמטרים:
  - הראשון הוא המקדם של $x^2$ (בדוגמה לעיל - a). ניתן להניח כי מקדם זה שונה מ-0.
  - השני הוא המקדם של x (בדוגמה לעיל - b).
  - השלישי הוא המקדם החופשי (בדוגמה לעיל - c).
- בכל מקרה הפונקציה תחזיר שני ערכים:
  - אם קיימים למשוואה שני פתרונות, ערכי ההחזרה הם שני הפתרונות הללו (ללא חשיבות לסדר).
  - אם קיים למשוואה פתרון יחיד, ערך ההחזרה הראשון הוא פתרון המשוואה והערך השני הוא <u>None</u>.
  - אם לא קיימים פתרונות למשוואה, שני ערכי ההחזרה של הפונקציה הם <u>None</u>.
- לדוגמה, קריאה לפונקציה עם הערכים:
  - (a,b,c) = (1, 1.5, -1) מייצגת את המשוואה $x^2 + 1.5x - 1 = 0$ אשר שני פתרונותיה הם (0.5) ו - (-2) ולכן ערכי ההחזרה של הפונקציה יהיו (0.5) ו - (-2) (או (-2) ו - (0.5)).
  - (a,b,c) = (1, -8, 16) מייצגת את המשוואה $x^2 - 8x + 16 = 0$ אשר לה פתרון יחיד 4 ולכן ערכי ההחזרה של הפונקציה יהיו 4 ו - None.
  - (a,b,c) = (1, -2, 34.5) מייצגת את המשוואה $x^2 - 2x + 34.5 = 0$ אשר לה אין פתרונות ממשיים ולכן ערכי ההחזרה של הפונקציה יהיו שניהם - None.
  - ניתן לייבא את הספריה math לצורך פתרון שאלה זו.
- הנחיות לפתרון משוואה ריבועית ניתן למצוא בויקיפדיה <u>כאן</u>.
- ניתן להניח כי הקלט לפונקציות הוא תקין.

**5. קבלת קלט מהמשתמש**:

פיטר פארקר בונה בעצמו חליפה משוכללת עבור האלטר אגו שלו כספיידרמן. הוא מעוניין להכניס את יכולת פתירת המשוואות הריבועיות לחליפה שלו, אך הוא מעוניין לשכלל את התכנית שלו כך שתכלול ממשק משתמש. כתבו פונקציה המבקשת מהמשתמש מקדמי משוואה ריבועית a, b ו - c ומדפיסה למסך את פתרונות המשוואה.

<u>הנחיות</u>:

- את הפונקציה יש לממש באותו הקובץ של הסעיף הקודם (ששמו quadratic_equation.py).
- שם הפונקציה הוא quadratic_equation_user_input.
- הפונקציה לא מקבלת פרמטרים.
- לא ניתן להניח ש - $a \neq 0$ .
- לאחר קריאה לפונקציה, הפונקציה תדפיס למסך את ההודעה הבאה (כולל תו הרווח המופיע בסוף המשפט לאחר הנקודתיים, ללא הגרשיים):

"Insert coefficients a, b, and c: "

- הפונקציה תמתין לקלט של שלושה מספרים מופרדים ברווח.
- לאחר הכנסת שלושת המספרים תדפיס הפונקציה הודעה למסך בהתאם להוראות שלהלן:
  - אם ערכו של a הוא אפס, הפונקציה תדפיס את ההודעה:

The parameter 'a' may not equal 0

  - אם קיימים למשוואה שני פתרונות, הפונקציה תדפיס את ההודעה:

The equation has 2 solutions: **X** and **Y**

כאשר ערכם של **X** ו - **Y** מוחלף בשני פתרונות המשוואה (אין חשיבות לסדר ההדפסה, כלומר לא משנה אם **X** מופיע בהדפסה לפני **Y**).

  - אם קיים למשוואה פתרון יחיד, הפונקציה תדפיס:

The equation has 1 solution: **X**

כאשר ערכו של **X** מוחלף בפתרון המשוואה.

  - אם לא קיימים פתרונות ממשיים למשוואה, הפונקציה תדפיס:

The equation has no solutions

- דוגמאות להרצה (קלט משתמש מסומן בצבע ירוק, הודעות המתחילות בתו # הן הסברים לצורך הבהרה ולא ייראו על המסך):
  - דוגמה 1:

Insert coefficients a, b, and c: 1 -8 15 *# User presses Enter*

*# Function outputs either:*

The equation has 2 solutions: 3.0 and 5.0

*# or:*

The equation has 2 solutions: 5.0 and 3.0

- דוגמה 2:

Insert coefficients a, b, and c: 1 -8 16 *# User presses Enter*

*# Function outputs*

The equation has 1 solution: 4.0

- דוגמה 3:

Insert coefficients a, b, and c: 1 1 1 *# User presses Enter*

*# Function outputs*

The equation has no solutions

- דוגמה 4:

Insert coefficients a, b, and c: 0 1 1 *# User presses Enter*

*# Function outputs*

The parameter 'a' may not equal 0

  - בפתרון סעיף זה יש לעשות שימוש בפונקציה מהסעיף הקודם.

**6. תנאים מקוננים**:

פיטר קווירל נחטף על-ידי חייזרים בתור ילד ולא הגיע לשיעור של פתרון משוואות ריבועיות, אך הוא הספיק ללמוד איך לפתור בעיות בגאומטריה ומנסה לחשב את השטחים של הצורות: עיגול, מלבן ומשולש שווה צלעות. כתבו פונקציה המבקשת מהמשתמש לבחור את הצורה שיש לחשב את שטחה, ואז מבקשת ממנו את הקלטים המתאימים לצורך חישוב השטח ומחזירה את השטח של הצורה המבוקשת על פי הנתונים.

<u>הנחיות</u> :

- את הפונקציה יש לממש בקובץ בשם shapes.py.
- שם הפונקציה הוא shape_area.
- קריאה לפונקציה תדפיס למסך את ההודעה (כולל תו הרווח המופיע בסוף המשפט לאחר הנקודתיים, ללא הגרשיים):

"Choose shape (1=circle, 2=rectangle, 3=triangle): "

- אם המספר המתקבל שונה מ 1, 2 או 3 על הפונקציה להחזיר את הערך None.
- עבור הקלט:
  - 1 - תחשב הפונקציה את שטחו של עיגול.
  - 2 - תחשב הפונקציה את שטחו של מלבן.
  - 3 - תחשב הפונקציה את שטחו של משולש שווה צלעות.
- עבור כל אחת מהצורות תמתין הפונקציה (ללא הדפסה נוספת למסך) למספר הקלטים המתאים בשורות נפרדות, על מנת לחשב את שטח הצורה הרצויה ואז תחזיר את שטחה.
  - עיגול: קלט יחיד - רדיוס.
    - קלט יחיד, כלומר קריאה אחת נוספת לקלט מהמשתמש
    - שטח עיגול שרדיוסו r, הוא $\pi \cdot r^2$ .
    - תזכורת - כדי להשתמש בקבוע π יש לייבא את הספרייה המתמטית של פיית'ון על ידי הוספת הפקודה import math לתכנית. הפקודה צריכה להיות מוספת במקום כלשהו לפני השימוש בה אך נהוג לשים אותה בראש התכנית. לאחר ייבוא הספרייה המתמטית, ניתן להשתמש בקבוע פאי על ידי math.pi .
  - מלבן: שני קלטים - אורך שתי צלעות המלבן.
    - שטח מלבן שאורך שתי צלעותיו הוא a ו - b הוא a * b.
  - משולש שווה צלעות: קלט בודד - אורך צלע המשולש.
    - שטח של משולש שווה צלעות שאורך צלעו a הוא $\frac{\sqrt{3}}{4}a^2$.

- דוגמאות להרצה (קלט משתמש מסומן בצבע ירוק, הודעות המתחילות בתו # הן הסברים לצורך הבהרה ולא ייראו על המסך):
  - דוגמה 1:

Choose shape (1=circle, 2=rectangle, 3=triangle): 1

*# User presses Enter (choice of a circle)*

5.3 *# User presses Enter (radius = 5.3)*

*# Function returns: 88.24733763933729*

  - דוגמה 2:

Choose shape (1=circle, 2=rectangle, 3=triangle): 2

*# User presses Enter (choice of a rectangle)*

5 *# User presses Enter (first square's side = 5)*

6 *# User presses Enter (first square's side = 6)*

*# Function returns: 30.0*

  - דוגמה 3:

Choose shape (1=circle, 2=rectangle, 3=triangle): 3

*# User presses Enter (choice of a* triangle*)*

9 *# User presses Enter (triangle side = 9)*

*# Function returns: 35.074028853269766*

- <u>הערה:</u> טרם המימוש, תכננו את החלוקה הפנימית של התכנית ועשו שימוש בפונקציות עזר כדי לעשות את הקוד מודולרי וקל להבנה.

**7. פונקציה המחזירה ערך בוליאני**:

תאנוס מעוניין לאסוף את ששת אבני האינסוף כדי למחוק מחצית מהיצורים החיים ביקום. אחת מאבני האינסוף, אבן הנשמה, נמצאת בכוכב הלכת וורמיר, לו יש שינויים קיצוניים במזג האוויר. הוא חייב לבחור להגיע לוורמיר ביום הנכון ולכן הוא בוחר מדד סטטיסטי. תאנוס יגיע לוורמיר רק כאשר הטמפרטורה הייתה **גבוהה ממש** מ-X מעלות במשך שניים מתוך שלושה ימים רצופים.

כתבו פונקציה המקבלת ערך סף מסויים (מספרי) ומדידות טמפרטורה של שלושה ימים ומחזירה האם בשניים מתוך שלושת הימים הטמפרטורה הייתה גבוהה ממש מערך הסף.

<u>הנחיות</u>:

- את הפונקציה יש לממש בקובץ בשם temperature.py.
- שם הפונקציה הוא is_vormir_safe.
- הפונקציה מקבלת ארבעה פרמטרים
  - הראשון - טמפרטורת סף מסויימת.
  - השני עד הרביעי, מדידות טמפרטורה משלושה ימים.
- הפונקציה תחזיר ערך בוליאני שערכו True אם לפחות שתיים מהמדידות היו גבוהות מערך הסף.
- לדוגמא, הקריאה:

```
is_ vormir_safe(19, 22, 21, 20)
```

  - תחזיר את הערך True כיוון שטמפרטורת הסף היא 19, ובכל שלושת הימים הטמפרטורה הייתה גבוהה ממנה.
- הקריאה:

```
is_ vormir_safe(20, 22, 21, 20)
```

  - תחזיר את הערך True כיוון שטמפרטורת הסף היא 20, וביומיים הראשונים הטמפרטורה הייתה מעל 20 מעלות.
- הקריאה:

```
is_ vormir_safe(7, 5, -2, 11)
```

  - תחזיר את הערך False כיוון שטמפרטורת הסף היא 7, ורק ביום אחד הטמפרטורה הייתה גבוהה ממנו.

- בפתרון השאלה אין להשתמש בפונקציה sort.

**8. פונקציה לבדיקת פונקציה 3:**

בסופו של דבר נקבע שברוס באנר הוא הנוקם החזק ביותר, למורת רוחו של ת'ור. ת'ור חושד כי תהליך הבחירה שהציע סטיב רוג'רס אינו כשר, ומעוניין לבצע עליו בדיקה חוזרת.

כחלק מתהליך התכנות, יש לממש פונקציה הבודקת את פונקציה 3 שמומשה על ידכם, כאשר תהליך הבדיקה יתבצע באופן הבא: פונקציה זו תריץ את הפונקציה משאלה 3 על הקלטים הבאים, ותבדוק האם הפלטים שהחזירה הם הפלטים הצפויים.

- קריאה לפונקציה עם הערכים 17,1,6 (כלומר הראשון 17, השני 1 והשלישי 6) צריכה להחזירה שני ערכים, הראשון 17 והשני 1.
- קריאה לפונקציה עם הערכים 1,17,6 צריכה להחזירה שני ערכים, הראשון 17 והשני 1.
- קריאה לפונקציה עם הערכים 1,1,2 צריכה להחזירה שני ערכים, הראשון 2 והשני 1.
- קריאה לפונקציה עם שלושה ערכים לבחירתכם (נסו לחשוב על קלט תקין שבודק מקרה קצה)
- קריאה לפונקציה עם שלושה ערכים לבחירתכם (נסו לחשוב על קלט תקין שבודק מקרה קצה).

הפונקציה תריץ את הפונקציה משאלה 3 על חמשת הקלטים הנ"ל בסדר הנתון, ותבדוק את ערכי החזרה של הפונקציה לכל אחד מחמשת המקרים הנ"ל. לבסוף, הפונקציה תחזיר ערך בוליאני <u>יחיד</u> המתאר האם הפלטים שהתקבלו בחמשת המקרים תאמו את ערכי ההחזרה הצפויים למקרים אלה. כלומר, הפונקציה תחזיר True אם כל הפלטים יצאו כפי שאמורים או False אחרת.

<u>הנחיות</u>:

- את הפונקציה יש לממש באותו הקובץ של סעיף 3 (ששמו largest_and_smallest.py).
- שם הפונקציה הוא check_largest_and_smallest.
- הפונקציה לא מקבלת פרמטרים ומחזירה ערך בוליאני כפי שתואר לעיל.
- עליכם להסביר בהערה (#) בראש הקובץ largest_and_smallest.py למה בחרתם את שני הקלטים האחרונים שבדקתם.

## נהלי הגשה

הגישו קובץ zip ששמו הוא ex2.zip המכיל את כל הקבצים הדרושים לתרגיל:

1. calculate_mathematical_expression.py
2. largest_and_smallest.py
3. quadratic_equation.py
4. shapes.py
5. temperature.py

# בהצלחה!