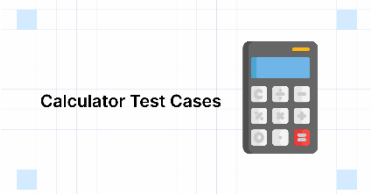
שיעורי בית

1. בהמשך לקוד שעשינו בכיתה-
https://github.com/pythonai170624/22.08.2024/blob/master/calculator.py
https://github.com/pythonai170624/22.08.2024/blob/master/test_calculator.py

a. הוסף למחשבון פונקצית power המקבלת שני פרמטרים int: מספר וחזקה, ומחזירה את המספר באותה החזקה. לדוגמא עבור 2 ו- 4 יחזור 16 ← כי 2 בחזקת 4 זה 16
b. הוסף למחשבון פונקצית sqrt המקבלת מספר, ומחזירה את השורש שלו. לדוגמא עבור 9 יחזור המספר 3 ← כי שורש 9 זה 3. יש להשתמש ב- math.sqrt (import math)
c. הוסף למחשבון פונקצית is_prime המקבלת מספר, ומחזירה אמת אם המספר ראשוני ושקר אם לא (שים לב ש- 1 הוא לא ראשוני, 0 ושלילי לא ראשוני)
הוסף למחשבון פונקצית factorial המקבלת מספר, ומחזירה את העצרת שלו.
לדוגמא עבור 3 יחזור 6 ← כי 1 * 2 * 3 = 6
לדוגמא עבור 5 יחזור 120 ← כי 1 * 2 * 3 * 4 * 5 = 120
עבור 0 ו- 1 יחזור 1,
<u>עבור מספר שלילי</u> יש לעלות **ValueError**. רמז: raise ValueError
בטקסט של השגיאה יש לרשום: "factorial not defined for negative values"
רמז: raise ValueError("factorial not defined for negative values")
(ראה קוד מהשיעור).

טסטים – כל טסט צריך להיות בפונקציה <u>נפרדת</u>. בדוק שכל הטסטים עברו בהצלחה!

d. הוסף פונקציית טסט הבודקת את פונקצית power, בטסט שלח את המספר 2 ו-4 לפונקציית power ובצע assert לבדוק שחזרה התוצאה 16 מהפונקציה (כמו בשיעור)
e. כמו בסעיף הקודם, בדוק בטסט שב- power עבור 3 ו- 2 חזר 9 (assert)
f. בדוק בטסט שב- power עבור 9 ו- 0 חזר 1 (השתמש ב assert)
g. בדוק בטסט שב- sqrt עבור 25 חזר 5 (השתמש ב assert)
h. בדוק בטסט שב- sqrt עבור **מינוס 5** התרחשה **שגיאה**:
ValueError: math domain error.
רמז: with pytest.raises(ValueError) (ראה קוד מהשיעור)
i. בדוק בטסט שב- is_prime עבור 1 חזר False (השתמש ב assert)
j. בדוק בטסט שב- is_prime עבור 2 חזר True (השתמש ב assert)
k. בדוק בטסט שב- is_prime עבור 16 חזר False (השתמש ב assert)
l. בדוק בטסט שב- is_prime עבור **מינוס 3** חזר False (השתמש ב assert)
m. בדוק בטסט שב- is_prime עבור 0 חזר False (השתמש ב assert)
n. בדוק בטסט שב- factorial עבור 4 חזר 24 (השתמש ב assert)
o. בדוק בטסט שב- factorial עבור 0 חזר 1 (השתמש ב assert)
p. בדוק בטסט שב- factorial עבור 1 חזר 1 (השתמש ב assert)
q. בדוק בטסט שב- factorial עבור 5 חזר 120 (השתמש ב assert)
r. בדוק בטסט שב- factorial עבור **מינוס 3** התרחשה **שגיאה**:
ValueError: factorial not defined for negative values.
רמז: with pytest.raises(ValueError) (ראה קוד מהשיעור)



המשך בעמוד הבא ...

2. <u>בונוס/רשות:</u>**

אבן נייר ומספרים

חוקי המשחק "אבן, נייר ומספריים" הם פשוטים מאוד:

1. **המשחק**:
   - כל אחד מהשחקנים בוחר אחת מהאופציות: אבן, נייר או מספריים.
   - הבחירה נעשית בו-זמנית כך שאף שחקן לא יודע מה השחקן השני בחר לפני שהוא עצמו בוחר.
2. **כללים**:
   - **אבן מנצחת מספריים** :האבן שוברת את המספריים.
   - **מספריים מנצחים נייר** :המספריים חותכים את הנייר.
   - **נייר מנצח אבן** :הנייר עוטף את האבן.
3. **תוצאה**:
   - אם שני השחקנים בוחרים את אותו דבר (למשל, שניהם בוחרים אבן), המשחק נגמר בתיקו.
   - אם אחד מהשחקנים בוחר אפשרות שמנצחת את זו של השחקן השני לפי הכללים המוזכרים, הוא מנצח.

<u>צור קובץ game.py עם הפונקציות הבאות:</u>

- פונקציית check_validity המקבלת כפרמטר מחרוזת str ומחזירה מספר int (או שגיאה):
  - אם התקבלה המחרוזת rock יחזור 2 (rock להיות באותיות גדולות/קטנות)
  - אם התקבל scissors יחזור 1 (scissors להיות באותיות גדולות/קטנות)
  - אם התקבלה המחרוזת paper יחזור 0 (paper להיות באותיות גדולות/קטנות)
  - עבור כל מחרוזת אחרת שהתקבלה יש לעלות **ValueError**.
    רמז: raise ValueError("illegal choice")
- פונקציית check_winner:
  - מקבלת כפרמטר שני מספרים int: player1_choice, player2_choice
  - מחזירה 1 אם השחקן הראשון ניצח, 2 אם השני ניצח ו- 0 אם תיקו
  - אם אחד הפרמטרים הוא לא 0, 1, 2 - יש לעלות **ValueError**.
    רמז: raise ValueError("illegal game option")
- פונקצית play_game המנהלת את המשחק. מציגה הודעות לשחקנים, קולטת את בחירת השחקנים, מפעילה את 2 הפונקציות האחרות. מדפיסה את תוצאת המשחק

<u>צור קובץ- main.py והפעל בו את המשחק, באמצעות פונקציית game.play_game</u>

<u>טסטים</u>

כתוב טסטים לפונקצית check_validity:

- בדוק <u>שכל מחרוזת</u> מחזירה את המספר הנכון (כל מחרוזת בטסט נפרד)
- הוסף טסטים לבדיקות אותיות גדולות וגם קטנות, עבור כל אפשרות ← ובדוק שחזר המספר הנכון. (כל מחרוזת בטסט נפרד).
- בדוק שמחרוזת לא חוקית גוררת שגיאה. רמז: with pytest.raises(ValueError)

המשך בעמוד הבא ...

כתוב טסטים לפונקצית check_winner::

- בדוק שכל אופציה מחזירה את המנצח/תיקו בצורה נכונה (כל אופציה בטסט נפרד)
  - **שחקן 1 בוחר אבן, שחקן 2 בוחר אבן** תיקו
  - **שחקן 1 בוחר אבן, שחקן 2 בוחר נייר** שחקן 2 מנצח
  - **שחקן 1 בוחר אבן, שחקן 2 בוחר מספריים** שחקן 1 מנצח
  - **שחקן 1 בוחר נייר, שחקן 2 בוחר אבן** שחקן 1 מנצח
  - **שחקן 1 בוחר נייר, שחקן 2 בוחר נייר** תיקו
  - **שחקן 1 בוחר נייר, שחקן 2 בוחר מספריים** שחקן 2 מנצח
  - **שחקן 1 בוחר מספריים, שחקן 2 בוחר אבן** שחקן 2 מנצח
  - **שחקן 1 בוחר מספריים, שחקן 2 בוחר נייר** שחקן 1 מנצח
  - **שחקן 1 בוחר מספריים, שחקן 2 בוחר מספריים** תיקו
- בדוק שאפשרות לא חוקית גוררת שגיאה. . רמז: with pytest.raises(ValueError)

בהצלחה!

את שיעורי הבית יש לשלוח ל- pythonai170624+HW13@gmail.com