

פייתון – מטלה 2

משקל המטלה = 4 נקודות. כל שאלה = נקודה אחת. אפשר לענות על חלק מהשאלות ולקבל ניקוד חלקי.

שאלה 1: חיפוש כתובות דואל בקובץ

כתבו תוכנית, המקבלת שם של קובץ טקסט, ומחפשת בו כתובות דואל.

התוכנית צריכה להדפיס שתי רשימות:

* רשימת כל הכתובות **התקינות** שהתגלו בקובץ. הגדרה של כתובת דואל תקינה נמצאת כאן:

https://help.xmatters.com/ondemand/trial/valid_email_format.htm

* רשימת כל הכתובות **הלא-תקינות** (נראות כמו כתובת דואל אבל עוברות על כלל אחד או יותר מהרשימה למעלה).

רמז: השתמשו בביטויים רגולריים.

שאלה 2: קשטן שזוכר את הקלט האחרון

כתבו קשטן בשם @lastcall, המקשט פונקציה כלשהי עם פרמטר אחד. הקשטן בודק האם הקלט הנוכחי זהה לקלט קודם כלשהו. אם לא – הוא מריץ את הפונקציה כרגיל. אם כן – הוא לא מריץ את הפונקציה, אלא כותב הודעה מתאימה. דוגמה:

```
@lastcall
```

```
def f(x: int):
```

```
    return x**2
```

```
f(2) # returns 4
```

```
f(2) # prints "I already told you that the answer is 4!"
```

```
f(10) # returns 100
```

```
f(2) # prints "I already told you that the answer is 4!"
```

```
f(10) # prints "I already told you that the answer is 100!"
```

בדקו את הקשטן שלכם על כמה פונקציות עם פרמטרים מסוגים שונים.

יש להגיש קישור לגיטהאב, הכולל קוד + תיעוד + בדיקות (דוקטסט) + דוגמאות הרצה.



שאלה 3: מערך רב-ממדי

כתבו מבנה בשם List, שהוא זהה לרשימה של פייתון (list), אבל מאפשר לגשת לפריטים בתחביר של מערך רב-ממדי. לדוגמה:

```
mylist = List([
    [[1,2,3,33],[4,5,6,66]],
    [[7,8,9,99],[10,11,12,122]],
    [[13,14,15,155],[16,17,18,188]],
])
print(mylist[0,1,3])
```

אמור להדפיס 66 (שורה 0, מערך 1, איבר 3).

בכל שאר הדברים, המבנה אמור להתנהג כמו רשימה של פייתון, למשל:

```
print(mylist[0])
```

אמור להדפיס:

```
[[1,2,3,33],[4,5,6,66]]
```

[רמז: השתמשו בירושה ובהעמסת אופרטורים].

יש להגיש קישור לגיטהאב, הכולל קוד + תיעוד + בדיקות + דוגמאות הרצה.

שאלה 4: משחק תיכנות

בחרו שאלה אחת מהרשימה הבאה: <https://www.codingame.com/training>

בדרגת קושי **בינונית – Medium** ומעלה. פתרו אותה בפייתון.

יש להגיש קישור לפתרון שלכם באתר codingame + צילום מסך.

