

פייתון – מטלה 2

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

שאלה 1: כתיבת תיעוד למודול

כתבו תוכנית, המקבלת כקלט שם של קובץ פייתון המכיל מודול מסויים, ושם של קובץ פלט, ויוצרת קובץ HTML המתעד את המודול: שמות הפונקציות שיש במודל, והתיעוד של כל פונקציה. דוגמה לשימוש:

```
python doc_to_html.py mymodule.py mydoc.html
```

לאחר ביצוע הפקודה, הקובץ mydoc.html יכיל את תיעוד המודול שבקובץ mymodule.py.

אפשר להניח שהמודול mymodule.py תקין.

רמז: אין צורך לנתח ידנית את הטקסט שבקובץ mymodule.py. אפשר להשתמש בשיטות אחרות, כגון eval או import. אפשר לקבל רשימה של כל הפונקציות במודול ע"י הפקודה dir.

הדגימו את פעולת הפונקציה שלכם.

שאלה 2: קשטן שזוכר את הקלט האחרון

כתבו קשטן בשם @lastcall, המקשט פונקציה כלשהי עם פרמטר אחד. הקשטן בודק האם הקלט הנוכחי זהה לקלט הקודם. אם כן – הוא כותב הודעה מתאימה. אם לא – הוא מריץ את הפונקציה כרגיל. 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(2) # prints "I already told you that the answer is 4!"
```

```
f(3) # returns 9
```

```
f(3) # prints "I already told you that the answer is 9!"
```

בדקו את הקשטן שלכם על כמה פונקציות.



שאלה 3: ייצור תת-קבוצות

כתבו פונקציה בשם `bounded_subsets`, המקבלת כקלט קבוצה S של מספרים חיוביים, ומספר חיובי כלשהו C , ומייצרת (ע"י `yield`) את כל התת-קבוצות של S , שסכומן לכל היותר C . הפתרון צריך להיות יעיל ככל האפשר. בפרט: אין לייצר את כל הקבוצות ואז לבדוק את הסכום, אלא לייצר מלכתחילה רק את הקבוצות שסכומן לכל היותר C . דוגמה:

for s in `bounded_subsets([1,2,3], 4)`:

```
print(s) # prints: [], [1], [2], [3], [1,2], [1,3].
```

בדקו הדגימו את הפונקציה שלכם.

שאלה 4: משחק תיכנות

בחרו שאלה נוספת כלשהי מאתר `Coding Game`, שונה מהקודמת:

<https://www.codinggame.com>

ופתרו אותה בעזרת פייתון.

יש להגיש קישור לדף של הפתרון שלכם באתר + צילום מסך.

