

מטלת פייתון – תבניות עיצוב

אפשר לענות על חלק מהשאלות ולקבל ניקוד חלקי. יש להשתמש בדוקטסט לצורך בדיקות.

שאלה 1: ייצור תת-קבוצות (2 נק')

כתבו איטרטור או גנרטור (לבחירתכם) בשם `bounded_subsets`. הוא מקבל כקלט קבוצה S של מספרים חיוביים, ומספר חיובי C , ומייצר סדרה של כל התת-קבוצות של S , שסכומן לכל היותר C .

הפתרון צריך להיות **יעיל**: אין לייצר את כל הקבוצות ואז לבדוק את הסכום, אלא לייצר מלכתחילה רק את הקבוצות שסכומן לכל היותר C . כמו כן, אין לייצר רשימה של כל התת-קבוצות הרלבנטיות, אלא איטרטור או גנרטור. דוגמאות הרצה:

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

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

```
for s in bounded_subsets(range(100,200), 203):
```

```
    print(s) # prints: [], [100], [101], [102], ..., [199], [100,101], [100,102], [100,103], [101,102]
```

הראו שהפונקציה שלכם אכן פותרת את הבעיה באופן יעיל.

שאלה 2: דגם-עיצוב אסטרטגיה (2 נק')

הגירסה הסופית של הקוד שהוצג בשיעור נמצאת בתיקיית הקוד, בקובץ בשם:

50-flyweight.py

כתבו מערכת דומה עבור בעיה אלגוריתמית אחרת כלשהי (לדוגמה: [בעיית הסוכן הנוסע](#)). המערכת שלכם צריכה לטפל באופן יעיל, ובלי שיכפול קוד, ב:

- לפחות שני סוגי קלט (לדוגמה: מרחקים בלבד, או מרחקים + שמות של ערים).
- לפחות שני סוגי פלט (לדוגמה: המסלול כולו, או רק אורך המסלול).
- לפחות שני אלגוריתמים. אפשר לבחור אלגוריתמים פשוטים וקצרים – לא דווקא אופטימליים.

השתמשו בדגמי-העיצוב שלמדנו בשיעור.

שאלה 3: משחק תכנות (1 נק')

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

בדרגת קושי **קשה – Hard** לפחות. פתרו אותה בפייתון.

יש להגיש קישור לפתרון שלכם באתר `codingame` + צילום מסך.

