

רשימות – תרגיל



ממש את הפעולות
בשאלות הבאות
במחלקה בשם
ListAddRemove

בכונן T בתיקייה Ayelet/Data Structure תמצאו פרוייקט בשם ListExample. היעזרו בפרוייקט על מנת לכתוב ולבדוק את הפעולות הבאות

- בדיקה:** על מנת לבדוק את הפעולות יש לבנות רשימה ולזמן אותן מ main עם הרשימה שיצרתם.
- השתמשו בפעולה buildList מהפרוייקט על מנת לבנות רשימה
 - השמשו בפעולה toString של Node<T> על מנת להדפיס את הרשימה לפני ואחרי הפעולה (כפי שמדגם ב main בפרוייקט)

1. כתוב פעולה בשם isSeries המקבלת רשימת מספרים ממשיים ומחזירה true אם היא מהווה סידרה חשבונית אחרת יוחזר false. עבור רשימה ריקה (null) או רשימה בת חוליה בודדת יוחזר false
2. כתוב פעולה בשם sumNum המקבלת רשימת שלמים שאינה ריקה ומספר שלם num - `sumNum(Node<Integer> lst, int num)`
יש לעדכן את הרשימה באופן הבא:
אחרי כל איבר ברשימה יתווסף איבר חדש כך שסכום כל זוג מספרים יהיה שווה ל num.
לדוגמא: עבור הרשימה

6 → 50 → 2 → 10 null

num = 20 י תועדכן הרשימה להיות

6 → 14 → 50 → -30 → 2 → 18 → 16 → 4 null

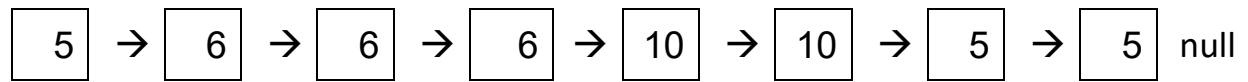
3. כתוב פעולה בשם evenOnly המקבלת רשימה של מספרים ומחזירה רשימה חדשה המכילה את כל המספרים הזוגיים ברשימה בסדר בו הם מופיעים.

4. כתוב פעולה בשם countSequense המקבלת רשימה ומספר שלם num - `countSequense(Node<Integer> lst, num)`
ומחזירה את מספר הרצפים של הערך ברשימה. לדוגמא עבור הרשימה:

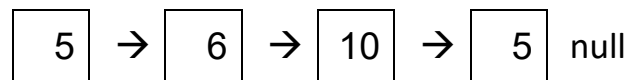
4 → 6 → 6 → 6 → 4 → 4 → 4 → 22 null

והמספר 4 תחזיר הפעולה 2

5. כתוב פעולה delSequense המקבלת רשימה שאינה ריקה. עבור כל רצף של מספרים זהים עוקבים יש להשאיר רק את הראשון ולבטל את שאר המספרים ברצף. לדוגמא רשימה



תעודכן להיות:



בהצלחה