

"תת-סדרה נגדיות" היא רצף של מספרים המתחילה במספר כלשהו ומסתיימים במספר הנגדי והשווה לו בערך המוחלט שלהם (כלומר מתחילה במספר חיובי ומסתיימים במספר השיליי המקביל לו או מתחילה במספר שלילי ומסתיימים במספר החיובי המקביל לו). הרצף נמצא בתחום סדרה של מספרים.

לדוגמה: בסדרת המספרים: 99, -17, 3, 1, -4, 5, 6, 4, -3, 7, 23, יש שתי תת-סדרות נגדיות:

i. 3, 1, -4, 5, 6, 4, -3

ii. -4, 5, 6, 4

נתונה שרשרת חוליות `lst` מtipוס `sh`, המכילה מספרים חיוביים ושליליים שאינם 0, ובולם שונים זה מהה (כלומר אין שני מספרים זהים בשרשרת).

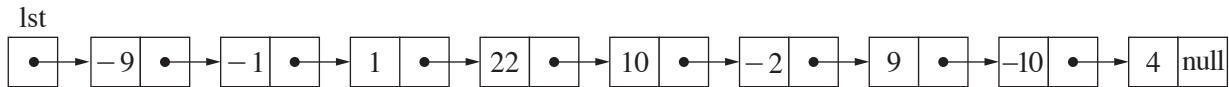
a. כתבו פעולה ששם `width` בשפת C# או `Width` בשפת Java או מקבלת את השרשרת `lst` ומספר שלם `num` (חיובי או שלילי) הופיע בשרשרת.

הפעולה תחזיר את אורך ה"תת-סדרה נגדיות" שהמספר `num` מתחילה או מסיים (האורך כולל את המספרים בקצוות). אם המספר הנגדי ל-`num` אינו מופיע בשרשרת, הפעולה תחזיר 1.

הערות: – חובה לשמור על השרשרת `lst`.

– אין להשתמש בסעיף זה במערך. פתרון הכלול שימוש במערך לא יזכה בנקודות.

דוגמאות: עבור השרשרת `lst` שלפניכם:



עבור `num = 9` הפעולה תחזיר 7 (התת-סדרה: -9, -1, 1, 22, 10, -2, 9).

עבור `num = -1` הפעולה תחזיר 2 (התת-סדרה: -1, 1).

עבור `num = 22` הפעולה תחזיר 1 – (משום שהמספר הנגדי לו, -22, אינו מופיע בשרשרת).

b. כתבו פעולה ששם `longest` בשפת C# או `Longest` בשפת Java או מקבלת את השרשרת `lst`. הפעולה תחזיר את אורך ה"תת-סדרה נגדיות" הגדולה ביותר. אם אין בשרשרת שום "תת-סדרה נגדיות", הפעולה תחזיר 1.

הערות: – אפשר להשתמש בפעולה שכתבתם בסעיף א.

– בפעולה זו אין חובה לשמור על השרשרת `lst`.

– אין להשתמש בסעיף זה במערך. פתרון הכלול שימוש במערך לא יזכה בנקודות.

דוגמא: עבור השרשרת `lst` שבדוגמה לעיל הפעולה תחזיר 7.

הסביר: ה"תת-סדרה נגדיות" המתחילה במספר 9 – ומסתיימת במספר 9 מכילה שבעה מספרים והוא הגדולה ביותר.