

"תת־סדרה נגדית" היא רצף של מספרים המתחיל במספר כלשהו ומסתיים במספר הנגדי והשווה לו בערך המוחלט שלהם

(כלומר מתחיל במספר חיובי ומסתיים במספר השלילי המקביל לו או מתחיל במספר שלילי ומסתיים במספר החיובי המקביל לו). הרצף נמצא בתוך סדרה של מספרים.

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

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

ii. $-4, 5, 6, 4$

נתונה שרשרת חוליות lst מטיפוס שלם, המכילה מספרים חיוביים ושליליים שאינם 0, וכולם שונים זה מזה (כלומר אין שני מספרים זהים בשרשרת).

א. כתבו פעולה ששמה width בשפת Java או Width בשפת C# המקבלת את השרשרת lst ומספר שלם num (חיובי או שלילי) המופיע בשרשרת.

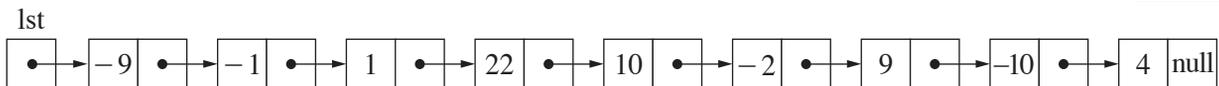
הפעולה תחזיר את אורך ה"תת־סדרה נגדית" שהמספר num מתחיל או מסיים (האורך כולל את המספרים בקצוות).

אם המספר הנגדי ל־ num אינו מופיע בשרשרת, הפעולה תחזיר 1 - .

הערות: - חובה לשמור על השרשרת lst.

- אין להשתמש בסעיף זה במערכ. פתרון הכולל שימוש במערכ לא יזוכה בנקודות.

דוגמאות: עבור השרשרת lst שלפניכם:



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

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

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

ב. כתבו פעולה ששמה longest בשפת Java או Longest בשפת C# המקבלת את השרשרת lst. הפעולה תחזיר את

אורך ה"תת־סדרה נגדית" הגדולה ביותר. אם אין בשרשרת שום "תת־סדרה נגדית", הפעולה תחזיר 1 - .

אפשר להשתמש בפעולה שכתבתם בסעיף א.

הערות: - בפעולה זו אין חובה לשמור על השרשרת lst.

- אין להשתמש בסעיף זה במערכ. פתרון הכולל שימוש במערכ לא יזוכה בנקודות.

לדוגמה: עבור השרשרת lst שבדוגמה לעיל הפעולה תחזיר 7.

הסבר: ה"תת־סדרה נגדית" המתחילה במספר 9 ומסתיימת במספר 9 מכילה שבעה מספרים והיא הגדולה ביותר.