רשימות מקושרות
שרשרת חוליות

שים לב! לכל תרגיל, יש לחשב את יעילות הפתרון

תרגיל 1

נתונה רשימה lst מסוג Node<Integer>.

כתוב פעולה בשם delete המקבלת כפרמטר את הרשימה lst ומספר שלם x ומוחקת את כל המופעים של x מהרשימה. בסיום נשארת הרשימה ללא מופעי x.

דוגמה:

הרשימה לפני המחיקה: lst [4 , 4 , 5 , 6 , 4 , 7 , 3 , 1 , 4 , 4 , 4 , 7 , 8 , 4] והמספר: x = 4

הרשימה אחרי המחיקה: lst [5 , 6 , 7 , 3 , 1 , 7 , 8]

תרגיל 2 (בגרות 2009)

רשימה תקרא **רשימה משולשת** אם היא לא ריקה, מספר האיברים שבה מתחלק ב- 3, האיברים בשליש הראשון שברשימה זהים לאיברים שבשליש השני ברשימה ולאיברים בשליש השלישי ברשימה.

לדוגמה: lst [2 , 5 , 7 , 9 , 2 , 5 , 7 , 9 , 2 , 5 , 7 , 9] הרשימה הינה **רשימה משולשת**

lst [2 , 5 , 7 , 9 , 2 , 5 , 7 , 9 , 2 , 5 , 9 , 7] הרשימה אינה **רשימה משולשת**

כתוב פעולה שתקבל כפרמטר רשימה lst ותחזיר אמת אם הרשימה היא **רשימה משולשת**, ושקר אחרת.

תרגיל 3 מיון הכנסה – Insertion Sort

א. כתוב פעולה בשם **הכנס בצורה ממוינת** – **InsertInSortedWay** שתקבל כפרמטר רשימה ממוינת lst של מספרים שלמים, ומספר שלם x ותכניס את x לרשימה כך שהרשימה תישאר ממויינת.
(המיון הוא בסדר עולה).

ב. כתוב פעולה בשם **מיון הכנסה - InsertionSort** שתקלוט סדרה של מספרים שלמים וחיוביים, עד לקבלת הזקיף -1
עבור כל מספר יש להכניסו לרשימה בצורה ממויינת.
הצג את הרשימה אחרי כל פעולה הכנסה.

תרגיל 4 מיזוג רשימות

נתונות שתי רשימות lst1 ו- lst2 המכילות מספרים שלמים. הרשימות ממוינות בסדר עולה.

כתוב פעולה בשם **מיזוג רשימות - mergLists** המקבלת כפרמטר את שתי הרשימות, ומחזירה רשימה שלישית המכילה את כל איברי lst1 ו- lst2 כשהם ממוינים בסדר עולה.
בסיום הפעולה, תישארנה הרשימות lst1 ו- lst2 ללא שינוי.

תרגיל 5

נתונות שתי רשימות של מספרים שלמים lst1 ו- lst2, בכל אחת מן הרשימות המספרים שונים זה מזה. נתון כי הרשימה lst2 ממוינת בסדר עולה.

א. כתוב פעולה בשם **חיסור רשימות** - **subtractLists**, המקבלת את שתי הרשימות ומחזירה רשימה חדשה המכילה את כל המספרים השלמים הנמצאים ב- lst1 אך לא נמצאים ב- lst2 .

ב. מה סדר גודל העבודה של הפעולה שכתבת? נמק.

תרגיל 6

בגרות תשנ"ח 1995

מקטע-משותף-מקסימלי לשתי רשימות הוא תת הרשימה הרצופה המקסימלית, המשותפת לשתיהן.
לדוגמה: עבור הרשימות lst1 ו- lst2 הבאות:

lst1: [1 , 5 , 6 , 3 , **4 , 8 , 9 , 5** , 4 , 3 , 6 , 7]

lst2: [5 , 4 , **4 , 8 , 9 , 5** , 6 , 3 , 6]

מהווה הרשימה lst3 רשימה המכילה את המקטע המשותף המקסימלי לשתיהן: lst3: [**4 , 8 , 9 , 5**]

נתונה חתימת הפעולה:

public static boolean ListsCompare (Node<Integer> lst1, Node<Integer> lst2,
Node<Integer> ps1, Node<Integer> pos2, **int** n)

הפעולה מקבלת כפרמטר שתי רשימות lst1 ו- lst2, הפניה לשני מקומות ברשימות – pos1 מפנה למקום ברשימה lst1 ו- pos2 מפנה למקום ברשימה lst2 ומספר שלם וחיובי n, ומחזירה *אמת* אם n האיברים החל מ- pos1 ברשימה lst1 זהים ל- n האיברים החל מ- pos2 ברשימה lst2, ו*שקר* אחרת.
אם באחת הרשימות, אין n איברים החל מהמקום שצוין, תחזיר הפעולה *שקר*.

כתוב פעולה בשם **מקטע משותף מקסימלי** – **mutualMaxSubNodes**, שתקבל כפרמטר שתי רשימות lst1 ו- lst2, ותחזיר רשימה המהווה את **המקטע המשותף המקסימלי** לשתיהן. ניתן להשתמש בפעולה הנתונה.

שים ♥: בבחינת הבגרות היה רשום: "הפעולה נתונה. אין צורך לממש אותה."
כדי להריץ את הפעולה במחשב יש לממש את הפעולה.

תרגיל 7

א. הרץ טבלת מעקב על הפעולה sod והרשימה lst שלהלן: lst: [5, 8, 2, 20, 7] ורשום מה מחזירה הפעולה.

ב. מה מבצעת הפעולה sod ? (רשום את טענת היציאה).

ג. הרץ טבלת מעקב על הפעולה mystery על הרשימות lst1 ו- lst2 שלהלן.

lst1: [5, 8, 2, 20, 7]
lst2: [ ]

ד. מה מבצעת הפעולה mistery ? (רשום את טענת היציאה).

```
public static Node<Integer>  sod (Node<Integer> lst1)
{
        Node<Integer> lst2 = null ;
        Node<Integer> pos = lst1;

        while (pos != null)
        {
                lst2 = new Node<Integer>( pos.getValue(), lst2);
                pos = pos.getNext();
        }
        return lst2;
}


public static void  mistery (Node<Integer> lst1, Node<Integer>lst2)
{
        if ( ! lst1.isEmpty())
        {
                Node<Integer> pos = lst1;
                lst2 = new Node<Integer>( pos.getValue(), lst2);
                lst1 = lst1.getNext();
                mystery (lst1, lst2);
        }
}
```