תת סדרה עולה

תבנית קוד לכתיבת הפתרון

```
public static  טיפוס הערך המוחזר  MethodName (Node<Integer> lst)
{
      int x, count;

      Node <Integer> pos = lst;
      while (pos != null)
      {
            x = pos.getValue();
            count = 1;

            pos = pos.getNext();

            while (pos != null  &&  pos. getValue()  ?  x)
            {


                  pos = pos.getNext();
            }
            if (count > 1)
                  _ _ _
      }

      return  ערך ההחזרה ;
}
```

אתחול משתנים או הרשימה המוחזרת

טיפול באיבר הראשון בתת הסדרה הנוכחית, ומעבר למספר הבא ברצף

טיפול באיבר הנוכחי בסדרה בהתאם לדרישות התרגיל

נגמרה תת הסדרה הנוכחית, טיפול בתוצאות בהתאם לנתוני השאלה

תרגיל 1

נתונה רשימה המכילה מספרים שלמים.

תת סדרה עולה זהו רצף של מספרים שכל אחד מהם גדול מהקודם לו.

יש לכתוב פעולה המקבלת כפרמטר רשימה של מספרים שלמים כזו, ומחזירה את אורך **תת הסדרה העולה** הארוכה ביותר.

לדוגמה: בעבור הרשימה שלהלן יוחזר המספר 8

[2 , 5 , 7 , 18 , 3 , 4 , 9 , 6 , 5 , 7 , 10 , -3 , -2 , -1 , 0 , 1 , 3 , 5 , 7]

```
public static int MaxLength (Node<Integer> lst)
{
      int max = 0;
      int x, count;

      Node <Integer> pos = lst;
      while (pos != null)
      {
            x = pos. getValue ();
            count = 1;

            pos = pos.getNext();

            while (pos != null  &&  pos.getValue () > x)
            {
                  count ++;
                  x = pos.getValue ();

                  pos = pos.getNext();
            }
            if (count > max)
                  max = count;
      }

      return max;
}
```

אתחול משתנים או הרשימה המוחזרת

ההשוואה נעשית בין שני איברים סמוכים, לכן יש לשמור את ערך x הקודם

נגמרה תת הסדרה הנוכחית, טיפול בתוצאות בהתאם לנתוני השאלה

תרגיל 2

נתונה רשימה המכילה מספרים שלמים.

תת סדרה עולה זהו רצף של מספרים שכל אחד מהם גדול מהקודם לו.

יש להחזיר את מיקומה של תחילתה של **תת הסדרה העולה** הארוכה ביותר

```
public static Node<Integer> LongestPlace (Node<Integer> lst)
{
    int max = 0;                    //    שומר את אורך תת הסדרה הארוכה ביותר שנמצאה עד כה
    Node<Integer> pMax;             //    שומר את מקום תחילת תת הרשימה הארוכה ביותר שנמצאה עד כה

    int x, count;

    Node <Integer> pos = lst;

    Node<Integer> p = pos;          //    שומר את מיקום תחילת תת הסדרה הנוכחית

    while (pos != null)
    {
        x = pos.getValue ();
        count = 1;

        pos = pos.getNext();

        while (pos != null  &&  pos.getInfo()  >  x)
        {
            count ++;
            x = pos.getValue ();

            pos = pos.getNext();
        }
        if (count > max)
        {
            max = count;
            pMax = p;
        }
        p = pos;
    }

    return pMax;
}
```

שמירת האורך המקסימלי ומיקום של איבר ראשון.
בכל מקרה (גם אם התנאי לא מתקיים, p מקבל את המיקום של תחילת תת הסדרה הבאה, ש- pos מפנה אליה.

תרגיל 3

כמו תרגיל 2, אבל הפעם יש להחזיר רשימה חדשה המכילה רק עותק של תת הסדרה העולה המקסימלית

```
public static Node<Integer> LongestSubList (Node<Integer> lst)
{
    int max = 0;                    //    שומר את אורך תת הסדרה הארוכה ביותר שנמצאה עד כה
    Node<Integer> pMax;             //    שומר את מקום תחילת תת הרשימה הארוכה ביותר שנמצאה עד כה

    int x, count;

    Node <Integer> pos = lst;

    Node<Integer> p = pos;          //    שומר את מיקום תחילת תת הסדרה הנוכחית

    while (pos != null)
    {
        x = pos.getValue ();
        count = 1;

        pos = pos.getNext();

        while (pos != null  &&  pos.getInfo()  >  x)
        {
            count ++;
            x = pos.getValue ();

            pos = pos.getNext();
        }
        if (count > max)
        {
            max = count;
            pMax = p;
        }
        p = pos;
    }

    //--- בניית והחזרה של תת-הסדרה המקסימלית ---
    Node<Integer> lst1 = new Node<Integer> (-1);      //-- חוליית דמה
    p = lst1;
    while (pMax != null  &&  max > 0)
    {
        p.setNext(new Node<Integer>(pMax.getValue(), p);
        p = p.getNext();
        max -- ;
        pMax = pMax.getNext();
    }
    return lst1.getNext();          //-- החזרת הרשימה החל מהאיבר שאחרי איבר הדמה
}
```

שמירת האורך המקסימלי ומיקום של איבר ראשון.
בכל מקרה (גם אם התנאי לא מתקיים, p מקבל את המיקום של תחילת תת הסדרה הבאה, ש- pos מפנה אליה.

p סיים את תפקידו בחיפוש תת הסדרה הארוכה ביותר, ולכן ניתן להשתמש בו עבור הרשימה החדשה