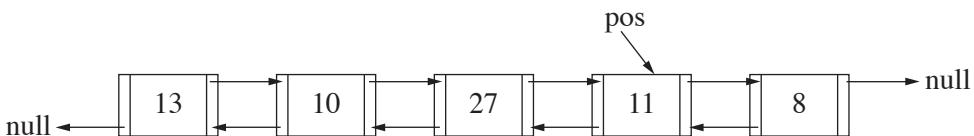


שים לב: לשאלה זו שני נוסחים:
נוסח אחד ב-Java בעמוד 8-9, ונוסח אחר ב-C# בעמ' 10-11.
עובד על פי השפה שלמדת.

לפונרים ב- Java

נגיד רשימה דו-כיוונית כאוסף סדר של חוליות מטיפוס `BinNode<Integer>` המוקשורות כך:
 לכל זוג חוליות `p1`, `p2` ברשימה, אם מתקיים `p1.getRight() == p2`, אז מתקיים גם
 `p1.getLeft() == p2`. ברשימה דו-כיוונית יש לפחות שתי חוליות.
כלומר: כל חוליה ברשימה — חוץ מהחוליה שבקצתה הימני של הרשימה והחוליה שבקצתה השמאלי
של הרשימה — מצביעה על החוליה שלפניה ועל החוליה שאחריה.
לפניך דוגמה לרשימה דו-כיוונית ומשתנה `pos` מטיפוס `BinNode<Integer>` המצביע על
חוליה כלשהי ברשימה דו-כיוונית.



הפעולה `firstLeft` מקבלת מצביע `pos` שונה מ-`null` מטיפוס `BinNode<Integer>` המצביע
על חוליה כלשהי ברשימה דו-כיוונית ומהזירה את החוליה השמאלית ביותר ברשימה.
הפעולה `firstRight` מקבלת מצביע `pos` שונה מ-`null` מטיפוס `BinNode<Integer>` המצביע
על חוליה כלשהי ברשימה דו-כיוונית ומהזירה את החוליה הימנית ביותר ברשימה.
א. לפניך שلد של הפעולה `.firstLeft`

העתק אותו למחרתך והשלם אותו, כך שהפעולה תבצע את הנדרש.

```

public static BinNode<Integer> firstLeft(BinNode<Integer> pos)
{
    while (_____)
        pos = _____;
    return _____;
}
  
```

.ב.

- לפניך הפעולה what(BinNode<Integer> pos) המקבלת מצביע לחוליה כלשהי ברשימה דו-כיוונית ומחזירה ערך בוליани. הרשימה הדו-כיוונית מכילה לפחות 3 חוליות.
- (1) עקב אחר ביצוע הפעולה בעבור המשתנה pos והרשימה שבדוגמה המוצגת בתחילת השאלה.

. pos, left, right, sum במקבב הראה את הרשימה הדו-כיוונית ואת ערכי המשתנים

```
public static boolean what(BinNode<Integer> pos)
```

```
{
```

```
    BinNode<Integer> left = firstLeft(pos);  
    BinNode<Integer> right = firstRight(pos);  
  
    int sum = left.getValue() + right.getValue();  
    left = left.getRight();  
    right = right.getLeft();
```

```
    while ((left != right) && (left.getRight() != right) &&  
           (left.getValue() + right.getValue() == sum))
```

```
{
```

```
    left = left.getRight();  
    right = right.getLeft();
```

```
}
```

```
if (left == right)
```

```
    return right.getValue() == sum;
```

```
if (left.getRight() == right)
```

```
    return left.getValue() + right.getValue() == sum;
```

```
return false;
```

```
}
```

- (2) קבע אם אפשר או אי אפשר להחליף את 3 השורות האחרונות שבפעולת — השורות המוקפות במסגרת — בהוראה:

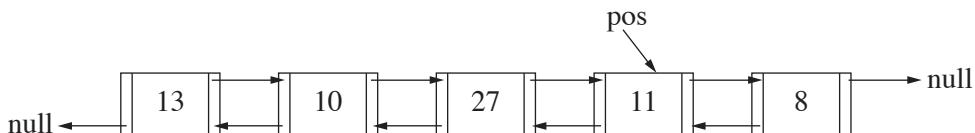
```
return left.getValue() + right.getValue() == sum;
```

نمך את קבועה.

נגיד רשימה דואכיאונית כאוסף סדר של חוליות מטיפוס `BinNode<int>` המוקשורות כך: לכל זוג חוליות `p1, p2` ברשימה, אם מתקיים `p1.GetRight() == p2`, אז מתקיים גם `p2.GetLeft() == p1`. ברשימה דואכיאונית יש לפחות שתי חוליות.

כלומר: כל חוליה ברשימה — חוץ מהחוליה שבקצה הימני של הרשימה והחוליה שבקצה השמאלי של הרשימה — מצביעה על החוליה שלפניה ועל החוליה שאחורה.

לפניך דוגמה לרשימה דואכיאונית ומשתנה `pos` מטיפוס `BinNode<int>` המצביע על חוליה כלשהי ברשימה דואכיאונית.



הפעולה `FirstLeft` מקבלת מצביע `pos` שונה מ-`null` מטיפוס `BinNode<int>` המצביע על חוליה כלשהי ברשימה דואכיאונית ומוחזירה את החוליה השמאלית ביותר ברשימה.

הפעולה `FirstRight` מקבלת מצביע `pos` שונה מ-`null` מטיפוס `BinNode<int>` המצביע על חוליה כלשהי ברשימה דואכיאונית ומוחזירה את החוליה הימנית ביותר ברשימה.

א. לפניך שلد של הפעולה `.FirstLeft`

העתק אותו למחברתך והשלם אותו, כך שהפעולה תבצע את הנדרש.

```

public static BinNode<int> FirstLeft(BinNode<int> pos)
{
    while (_____)
        pos = _____;
    return _____;
}
  
```

.ב.

- לפניך הפעולה What(BinNode<int> pos) המקבלת מצביע לחוליה כלשהי ברשימה דו-כיוונית ומחזירה ערך בוליאני. הרשימה הדו-כיוונית מכילה לפחות 3 חוליות.
- (1) עקב אחר ביצוע הפעולה עבור המשתנה pos והרשימה שבדוגמה המוצגת בתחלת השאלה.

. pos, left, right, sum

```
public static bool What(BinNode<int> pos)
{
    BinNode<int> left = FirstLeft(pos);
    BinNode<int> right = FirstRight(pos);

    int sum = left.GetValue() + right.GetValue();
    left = left.GetRight();
    right = right.GetLeft();

    while ((left != right) && (left.GetRight() != right) &&
           (left.GetValue() + right.GetValue() == sum))
    {
        left = left.GetRight();
        right = right.GetLeft();
    }
    if (left == right)
        return right.GetValue() == sum;

    if (left.GetRight() == right)
        return left.GetValue() + right.GetValue() == sum;
    return false;
}
```

קבע אם אפשר או אי אפשר להחליף את 3 השורות האחרונות שבפעולה — השורות המוקפות במסגרת — בהוראה:

return left.GetValue() + right.GetValue() == sum;

נק את קביעותן.