

# Node

## רשימה מקושרת

```
// רשימה מקושרת
// 400. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה כמה איברים ברשימה
public static int CountOfNodeNumbers(Node<int> node)
{
    int count = 0;
    while (node != null)
    {
        count++;
        node = node.GetNext();
    }
    return count;
}

// 401. כתוב פעולה המקבלת רשימה מסוג מספר שלם ומחזירה את סכום האיברים
public static int SumOfNodeNumbers(Node<int> node)
{
    int sum = 0;
    while (node != null)
    {
        sum = sum + node.GetValue();
        node = node.GetNext();
    }
    return sum;
}

// 402. נמצא NUM ומחזירה האם, NUM כתוב פעולה המקבלת רשימה מסוג מספר שלם ומספר שלם.
// ברשימה
public static bool NumberInTheNode(Node<int> node, int num)
{
    while (node != null)
    {
        if (node.GetValue() == num)
        {
            return true;
        }
        node = node.GetNext();
    }
    return false;
}

// 403. כתוב פעולה המקבלת רשימה מסוג מספר שלם ובודקת האם בכל הרשימה יש רק מספרים זוגיים
public static bool EvenNodeNumbers(Node<int> node)
{
    while (node != null)
    {
        if (node.GetValue() % 2 == 1)
        {
            return false;
        }
        node = node.GetNext();
    }
    return true;
}

// 404. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה את האיבר הגדול ביותר. (הרשימה אינה ריקה)
public static int TheBiggestNode(Node<int> node)
```

```

{
    int maxNode = node.GetValue();
    while (node != null)
    {
        if (node.GetValue() > maxNode)
        {
            maxNode = node.GetValue();
        }
        node = node.GetNext();
    }
    return maxNode;
}

```

// 405. ומחזירה האם סכום האיברים גדול, NUM, כתוב פעולה המקבלת רשימה מסוג מספר שלם ומספר שלם.  
 // מNUM.

```

public static bool NumberBiggerThenSumOfNode(Node<int> node, int num)
{
    int sum = 0;
    while (node != null)
    {
        sum = sum + node.GetValue();
        node = node.GetNext();
    }

    if (num > sum)
    {
        return true;
    }
    return false;
}

```

// 406. NUM ומחזירה האם סכום NUM כתוב פעולה המקבלת רשימה מסוג מספר שלם ועוד מספר שלם.  
 האיברים

// הראשונים גדול מסכום שאר האיברים.  
 // א.אפשר להניח שיש יותר איברים מ

```

public static bool IfNumberBiggerThenTheOtherNodes(Node<int> node, int num)
{
    int sum1 = 0;
    int sum2 = 0;
    for (int i = 0; i < num; i++)
    {
        sum1 = sum1 + node.GetValue();
        node = node.GetNext();
    }

    while (node != null)
    {
        // NUM הלולאה רצה לאחר
        sum2 = sum2 + node.GetValue();
        node = node.GetNext();
    }

    if (sum1 > sum2)
    {
        return true;
    }
    return false;
}

```

// 406. NUM ומחזירה האם סכום NUM כתוב פעולה המקבלת רשימה מסוג מספר שלם ועוד מספר שלם.  
 האיברים

```

// הראשונים גדול מסכום שאר האיברים
// (יכול להיות שיש פחות NUM ב.א. אפשר להניח שיש יותר איברים מ
public static bool IfNumberBiggerThenTheOtherNodes2(Node<int> node, int
num)
{
    int sum = 0;

    Node<int> pointer = node;

    int count = 0;
    while (pointer != null)
    {
        count++;
        pointer = pointer.GetNext();
    }

    if (count > num)
    {
        return IfNumberBiggerThenTheOtherNodes(node, num);
    }

    while (node != null)
    {
        sum = sum + node.GetValue();
        node = node.GetNext();
    }

    if (sum > 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

// 407. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה האם סכום החצי הראשון של הרשימה שווה לסכום החצי השני של הרשימה.  
 // א. מספר האיברים ברשימה הוא זוגי  
 // ב. מספר האיברים ברשימה הוא אי זוגי, ואז האיבר האמצעי לא נכלל בסכום  
 // ג. לא ידוע אם מספר האיברים או זוגי או אי זוגי (אם אי זוגי לדלג על האיבר האמצעי)  
 // ד. להשתמש רק במשתנה אחד בשביל הסכום

```

public static bool SumOfHalf(Node<int> node)
{
    int sum = 0;
    int count = 0;
    Node<int> pointer = node;
    while (pointer != null)
    {
        count++;
        pointer = pointer.GetNext();
    }

    if (count % 2 == 0)
    {
        for (int i = 0; i < count / 2; i++)
        {
            sum = sum + node.GetValue();
            node = node.GetNext();
        }
    }
}

```

```

        int sum1 = sum;
        sum = 0;
        for (int i = 0; i < count / 2; i++)
        {
            sum = sum + node.GetValue();
            node = node.GetNext();
        }

        if (sum == sum1)
        {
            return true;
        }
        return false;
    }

else
{
    for (int i = 0; i < count / 2; i++)
    {
        sum = sum + node.GetValue();
        node = node.GetNext();
    }

    node = node.GetNext();

    for (int i = 0; i < count / 2; i++)
    {
        sum = sum + node.GetValue();
        node = node.GetNext();
    }

    return true;
}
}

```

// 408. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה האם הרשימה ממוינת בסדר עולה.

```

public static bool AscendingNode(Node<int> node)
{
    if (node == null)
    {
        return true;
    }

    while (node.GetNext() != null)
    {
        if (node.GetValue() > node.GetNext().GetValue())
        {
            return false;
        }
        node = node.GetNext();
    }
    return true;
}

```

// 409. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה האם כל איבר הוא פי 2 מהאיבר הקודם.

```

public static bool TwiceNode(Node<int> node)
{
    if (node == null)
    {
        return true;
    }
}

```

```

while (node.GetNext() != null)
{
    if (node.GetValue() * 2 != node.GetNext().GetValue())
    {
        return false;
    }
    node = node.GetNext();
}
return true;
}

```

NUM ומחזירה את סכום NUM א. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומספר שלם. 410. // האיברים

קטן ממספר האיברים ברשימה NUM הראשונים. אפשר להניח ש

```

public static int SumOfFirstNode(Node<int> node, int num)
{
    Node<int> lst = node;
    int len = 0;
    while (lst != null)
    {
        len++;
        lst = lst.GetNext();
    }
    if (len < num)
    {
        return int.MinValue;
    }

    int sum = 0;
    for (int i = 0; i < num; i++)
    {
        sum = sum + node.GetValue();
        node = node.GetNext();
    }
    return sum;
}

```

410. ב. //

ומחזירה את הסכום הכי גדול, NUMBER ב. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומספר שלם. הוא 4, אז לחפש ברשימה את הרצף NUMBER בתוך הרשימה. (דוגמה, אם NUMBER של רצף המספרים) של 4 המספרים שהסכום שלהם הכי גדול, ולהחזיר את הסכום הזה) (שם לב שחייבים רצף של) מספרים ולא פחות, ושם NUMBER חובה להשתמש בפעולה שכתבת בסעיף א. (שם לב שחייבים רצף של) (לב לא לצאת מהרשימה) //

```

public static int SumOfBiggestSequence(Node<int> node, int num)
{
    int sum = 0;
    int maxSum = 0;
    Node<int> pointer = node;
    while (pointer != null)
    {
        sum = SumOfFirstNode(pointer, num);
        if (sum > maxSum)
        {
            maxSum = sum;
        }
        pointer = pointer.GetNext();
    }
    return maxSum;
}

```

// 411. כתוב פעולה המקבלת רשימה מסוג שלם, ומחזירה האם האיברים בחצי הראשון של הרשימה מסודרים בסדר

עולה, והאיברים בחצי השני של הרשימה מסודרים בסדר יורד. מספר האיברים ברשימה הוא זוגי.

```
public static bool AccordingNodeHalf(Node<int> node)
{
    int count = 0;
    Node<int> pointer = node;
    while (pointer != null)
    {
        count++;
        pointer = pointer.GetNext();
    }

    int half = count / 2;
    Node<int> pointer2 = node;
    for (int i = 1; i < half; i++)
    {
        if (pointer2.GetValue() > pointer2.GetNext().GetValue())
        {
            return false;
        }
        pointer2 = pointer2.GetNext();
    }
    pointer2 = pointer2.GetNext();

    for (int i = 1; i < half; i++)
    {
        if (pointer2.GetValue() < pointer2.GetNext().GetValue())
        {
            return false;
        }
        pointer2 = pointer2.GetNext();
    }
    return true;
}
```

// 412. א.

כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומדפיסה את האיברים הרשימה

א. מההתחלה לסוף

```
public static void PrintNode(Node<int> node)
{
    if (node == null)
    {
        return;
    }
    Console.WriteLine(node.GetValue());
    PrintNode2(node.GetNext());
}
```

// 412. ב.

כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומדפיסה את האיברים הרשימה

ב. מהסוף להתחלה

```
public static void PrintNode2(Node<int> node)
{
    if (node == null)
    {
        return;
    }
    PrintNode2(node.GetNext());
    Console.WriteLine(node.GetValue());
}
```

```
}
```

```
// 413. א.
```

```
// כתוב פעולה המקבלת רשימה מסוג מספר שלם ומחזירה:
```

```
// אם מספר האיברים ברשימה היא אי זוגית, אז מחזירה את האיבר האמצעי/
```

```
// אם מספר האיברים ברשימה היא זוגית, אז מחזירה את הממוצע של 2 האיברים האמצעיים/
```

```
//0. אם הרשימה ריקה, מחזירה 0.
```

```
//א. ניתן להשתמש בכמה לולאות שצריך/
```

```
public static double MiddleNode(Node<int> node)
```

```
{
```

```
    if (node == null)
```

```
    {
```

```
        return 0;
```

```
    }
```

```
    if (node == null)
```

```
    {
```

```
        return 0;
```

```
    }
```

```
    int count = 0;
```

```
    Node<int> pointer = node;
```

```
    while (pointer != null)
```

```
    {
```

```
        count++;
```

```
        pointer = pointer.GetNext();
```

```
    }
```

```
    if (count % 2 == 1)
```

```
    {
```

```
        int middle = count / 2;
```

```
        for (int i = 0; i < middle; i++)
```

```
        {
```

```
            node = node.GetNext();
```

```
        }
```

```
        return node.GetValue();
```

```
    }
```

```
    else
```

```
    {
```

```
        int middle = count / 2;
```

```
        for (int i = 0; i < middle - 1; i++)
```

```
        {
```

```
            node = node.GetNext();
```

```
        }
```

```
        return (node.GetValue() + node.GetNext().GetValue()) / 2.0;
```

```
    }
```

```
}
```

```
// 413. ב.
```

```
// כתוב פעולה המקבלת רשימה מסוג מספר שלם ומחזירה
```

```
// אם מספר האיברים ברשימה היא אי זוגית, אז מחזירה את האיבר האמצעי/
```

```
// אם מספר האיברים ברשימה היא זוגית, אז מחזירה את הממוצע של 2 האיברים האמצעיים/
```

```
//אם הרשימה ריקה, מחזירה 0.
```

```
// ב. ניתן להשתמש רק בלולאה אחת
```

```
public static double MiddleNode2(Node<int> node)
```

```
{
```

```
    if (node == null)
```

```
    {
```

```
        return 0;
```

```
    }
```

```

Node<int> pointer1 = node;
Node<int> pointer2 = node;
while (pointer2.GetNext() != null && pointer2.GetNext().GetNext() !=
null)
{
    pointer1 = pointer1.GetNext();
    pointer2 = pointer2.GetNext().GetNext();
}

if (pointer2.GetNext() == null)
{
    return pointer1.GetValue();
}
return (pointer1.GetValue() + pointer1.GetNext().GetValue()) / 2.0;
}

```

// 414. כתוב פעולה המקבלת רשימה מסוג מספר שלם שבו כל איבר הוא ספרה (בין 0 ל 9 כולל) הרשימה מייצגת

// 'מספר שלם. כאשר ספרת האחדות נמצאת בתחילת הרשימה, העשרות האיבר השני וכו'.

// יש להחזיר את המספר כמספר שלם.

```

public static int NumberBetweenZeroToNine(Node<int> node)
{
    int number = 0;
    while (node != null)
    {
        number = number * 10 + node.GetValue();
        node = node.GetNext();
    }
    return number;
}

```

// 415. א.

כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה האם האיברים בחצי הראשון, נמצאים גם בחצי השני. (מספר האיברים ברשימה זוגי).

א. האיברים מסודרים לפי אותו הסדר.

```

public static bool TheSameNumbers(Node<int> node)
{
    int count = 0;
    Node<int> pointer = node;
    while (pointer != null)
    {
        count++;
        pointer = pointer.GetNext();
    }

    int half = count / 2;
    Node<int> pointer2 = node;
    for (int i = 0; i < half; i++)
    {
        pointer2 = pointer2.GetNext();
    }
    for (int i = 0; i < half; i++)
    {
        if (pointer2.GetValue() != node.GetValue())
        {
            return false;
        }
        node = node.GetNext();
        pointer2 = pointer2.GetNext();
    }

    return true;
}

```



```
}
```

```
// 415. ב.
```

כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה האם האיברים בחצי הראשון, נמצאים גם בחצי השני. (מספר האיברים ברשימה זוגי)

ב. האיברים לא מסודרים לפי אותו הסדר

```
public static bool TheSameNumbers2(Node<int> node)
{
```

```
    int count = 0;
    Node<int> pointer = node;
    while (pointer != null)
    {
        count++;
        pointer = pointer.GetNext();
    }
```

```
    int half = count / 2;
    Node<int> pointer2 = node;
    for (int i = 0; i < half; i++)
    {
        Node<int> pointer3 = node;
        for (int j = 0; j < half; j++)
        {
            if (pointer2.GetValue() == pointer3.GetValue())
            {
                break;
            }
            pointer3 = pointer3.GetNext();
        }
        if (pointer3 == null)
        {
            return false;
        }
        pointer2 = pointer2.GetNext();
    }
    return true;
}
```

416. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה את המיקום שבו סכום האיברים לפני שווה לסכום האיברים אחרי (לא כולל אותו המקום) אם אין מקום כזה, יש להחזיר -1. יש לפחות 3 איברים ברשימה

```
public static int SumBeforeAndAfter(Node<int> node)
{
```

```
    int SumAll = 0;
    Node<int> ptr = node;
    while (ptr != null)
    {
        SumAll += ptr.GetValue();
        ptr = ptr.GetNext();
    }

    int index = 0;
    int SumBefore = node.GetValue();
    int SumAfter = SumAll - SumBefore;
    node = node.GetNext();
    while (node != null)
    {
        index++;
        SumAfter -= node.GetValue();
    }
```

```

        if (SumBefore == SumAfter)
        {
            return index;
        }
        SumBefore += node.GetValue();

        node = node.GetNext();
    }
    return -1;
}

```

כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה האם יש רצף מספרים שחוזר על עצמו. // 417. רצף של

```

//לפחות 2 מספרים)
public static bool SequenceAgain(Node<int> node)
{
    Node<int> pointer1 = node;
    Node<int> pointer2 = node.GetNext();
    while (pointer2 != null)
    {
        if (pointer1.GetValue() == pointer2.GetValue())
        {
            return true;
        }
        pointer1 = pointer1.GetNext();
        pointer2 = pointer2.GetNext();
    }
    return false;
}

```

## רשימה מקושרת – שינוי/בניית רשימה

רשימה מקושרת – שינוי\בניית רשימה  
 הפעולה תוציא מהרשימה את NUM, כתוב פעולה המקבלת רשימה מסוג מספר שלם ומספר שלם. // 430. NUM.

יכול להופיע פעם אחת, ולא בראש הרשימה NUM המספר  
 public static void RemoveNumber(Node<int> node, int num)  
 {  
 Node<int> pointer = node;  
 while (pointer.GetNext() != null)  
 {  
 if (pointer.GetNext().GetValue() == num)  
 {  
 pointer.SetNext(pointer.GetNext().GetNext());  
 return;  
 }  
 pointer = pointer.GetNext();  
 }  
 }  
}

הפעולה תוציא מהרשימה את NUM, כתוב פעולה המקבלת רשימה מסוג מספר שלם ומספר שלם. // 431. NUM.

יכול להופיע יותר מפעם אחת, ולא בראש הרשימה NUM המספר  
 public static void RemoveNumber2(Node<int> node, int num)  
 {  
 Node<int> pointer = node;  
 while (pointer.GetNext() != null)  
 {  
 if (pointer.GetNext().GetValue() == num)  
 {

```

        pointer.SetNext(pointer.GetNext().GetNext());
    }

    else
    {
        pointer = pointer.GetNext();
    }
}
}

```

// 432. ה-**NUM** הפעולה תכניס את **NUM**, כתוב פעולה המקבלת רשימה ממוינת מסוג מספר שלם ומספר שלם **NUM**, למקומו

// לא צריך להיכנס לראש הרשימה **NUM**. לפי המיון

```

public static void InsertNumber(Node<int> node, int num)
{
    Node<int> newNode = new Node<int>(num);
    Node<int> pointer = node;
    while (pointer.GetNext() != null)
    {
        if (pointer.GetNext().GetValue() > num)
        {
            newNode.SetNext(pointer.GetNext());
            pointer.SetNext(newNode);
            return;
        }
        pointer = pointer.GetNext();
    }
    pointer.SetNext(newNode);
}

```

// 433. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה שכפול של הרשימה

```

public static Node<int> DoubleNode(Node<int> node)
{
    Node<int> newHead = new Node<int>(node.GetValue());
    Node<int> pointer = newHead;
    Node<int> pointer2 = node.GetNext();
    while (pointer2 != null)
    {
        pointer.SetNext(new Node<int>(pointer2.GetValue()));
        pointer = pointer.GetNext();
        pointer2 = pointer2.GetNext();
    }
    return newHead;
}

```

// 434. כתוב פעולה המקבלת 2 רשימות מסוג מספר שלם, ומחזירה רשימה חדשה שבה כל איבר הוא סכום של 2

איברים מ-2 הרשימות. הסכום הוא לפי הסדר. מספר האיברים יכול להיות שונה

```

public static Node<int> SumOfTwoNodes(Node<int> node1, Node<int> node2)
{
    Node<int> newHead = new Node<int>(node1.GetValue() +
node2.GetValue());
    Node<int> pointer = newHead;
    Node<int> pointer1 = node1.GetNext();
    Node<int> pointer2 = node2.GetNext();
    while (pointer1 != null && pointer2 != null)
    {
        pointer.SetNext(new Node<int>(pointer1.GetValue() +
pointer2.GetValue()));
        pointer = pointer.GetNext();
    }
}

```

```

        pointer1 = pointer1.GetNext();
        pointer2 = pointer2.GetNext();
    }

    while (pointer1 != null)
    {
        pointer.SetNext(new Node<int>(pointer1.GetValue()));
        pointer = pointer.GetNext();
        pointer1 = pointer1.GetNext();
    }

    while (pointer2 != null)
    {
        pointer.SetNext(new Node<int>(pointer2.GetValue()));
        pointer = pointer.GetNext();
        pointer2 = pointer2.GetNext();
    }
    return newHead;
}

// 435. כתוב פעולה המקבלת רשימה מסוג שלם ומוחקת את האיבר הראשון ברשימה, הרשימה לא ריקה.
public static void RemoveFirstNode(Node<int> node)
{
    node.SetValue(node.GetNext().GetValue());
    node.SetNext(node.GetNext().GetNext());
}

// 436. כתוב פעולה המקבלת רשימה מסוג מספר שלם, ומחזירה רשימה חדשה שהסדר הוא הפוך.
// דוגמה: הרשימה לפני: 1 2 3 4 5
// הרשימה המוחזרת: 5 4 3 2 1
public static Node<int> ReverseNode(Node<int> node)
{
    Node<int> newHead = new Node<int>(node.GetValue());
    Node<int> pointer = node.GetNext();
    while (pointer != null)
    {
        Node<int> newNode = new Node<int>(pointer.GetValue());
        newNode.SetNext(newHead);
        newHead = newNode;
        pointer = pointer.GetNext();
    }
    return newHead;
}

// 437. כתוב פעולה המקבלת מספר ובונה רשימה כך שכל ספרה תהיה במקום ברשימה מהאחדות והלאה,
// לדוגמה,
// עבור המספר 123, יוחזר רשימה עם 3 בחוליה הראשונה, 2 בחוליה השנייה ו1 בחוליה השלישית.
public static Node<int> NumberToNode(int num)
{
    Node<int> newHead = new Node<int>(num % 10);
    num = num / 10;
    Node<int> pointer = newHead;
    while (num != 0)
    {
        pointer.SetNext(new Node<int>(num % 10));
        num = num / 10;
        pointer = pointer.GetNext();
    }
    return newHead;
}

```

// 438. כתוב פעולה המקבלת מספר ובונה רשימה כך שכל ספרה תהיה במקום ברשימה המספרה הכי משמעותית  
// עד האחדות, לדוגמה, עבור המספר 123, יוחזר רשימה עם 1 בחוליה הראשונה, 2 בחוליה השניה ו2 בחוליה השלישית.

```
public static Node<int> NumberToNode2(int num)
{
    Node<int> newHead = new Node<int>(num % 10);
    num = num / 10;
    Node<int> pointer = newHead;
    while (num != 0)
    {
        Node<int> newNode = new Node<int>(num % 10);
        newNode.SetNext(pointer);
        pointer = newNode;
        num = num / 10;
    }
    return pointer;
}
```