

# חוברת שאלות

## מבנה נתונים ותכנות מונחה עצמים

מתוך בחינות הבגרות  
בין השנים תשסו-תשעז

(שאלונים 899205, 899381)

### תוכן עניינים

<b>2</b>	<b>מבנה נתונים</b>
2	עצמים ומחלקות
4	רקורסיה
15	רשימה
21	מחסנית ותור
23	אוספים
31	עצים
<b>39</b>	<b>תכנות מונחה עצמים</b>
39	מעקב אחרי Main
39	המרות זיהוי שגיאות
39	ממשקים
39	מימוש תכנות מערכות

## מבנה נתונים

### עצמים ומחלקות

1. [בגרות תשסז] במסעדת "טעמים22" מנהלים מערכת ממוחשבת המרכזת מידע על המנות המוגשות בה.
 

**מנה במסעדה** מיוצגת על ידי: קוד המנה, שם המנה, ורשימת סוגי המזון המרכיבים את המנה (למשל בעבור סלט אבוקדו: אבוקדו, פלפל אדום, בצל סגול, מיץ לימון). בכל מנה יכולים להיות לכל היותר 15 סוגי מזון.

**תפריט במסעדה** מיוצג על ידי: שם התפריט, מערך בגודל של 22, שכל איבר בו הוא **מנה במסעדה**.

א. כתוב את כותרת המחלקה ואת התכונות בעבור כל אחד מהמחלקות: **מנה במסעדה** – **Course**, **תפריט במסעדה** – **Menu**.

ב. כתוב פעולה במחלקה **Menu** המקבלת סוג מזון אחד  $f$  החסר במטבח המסעדה. הפעולה תדפיס מבין הקודים של המנות הכלולות בתפריט, רק את הקודים של המנות שסוג המזון  $f$  אינו אחד מהמרכיבים שלהן. הנח כי עבור כל תכונה במחלקות **Course**, **Menu** הוגדרו פעולות **Get** ו-**Set**.
2. [בגרות תשעג] לאבן משחק יש שני צדדים. על צד אחד רשום אחד המספרים החד-ספרתיים בין 0 ל-6 (כולל), ועל הצד השני רשום אחד מהמספרים הדו-ספרתיים בין 10 ל-16 (כולל).
 

א. כתוב את כותרת המחלקה **BiStone**, המגדירה את אבן המשחק, ואת התכונות שלה.

ב. כתוב במחלקה **BiStone** פעולה בונה המקבלת שני מספרים, האחת חד-ספרתי והאחר דו-ספרתי בין 10 ל-16.

ג. כתוב במחלקה **BiStone** פעולה פנימית שתחזיר **true** אם המספר החד-ספרתי שבצד האחד של אבן המשחק שווה לספרת האחדות של המספר הדו-ספרתי שמופיע בצד השני של האבן, אחרת – **false**.

המחלקה **Stones** היא אוסף של כל אבני **BiStone** האפשריות. המחלקה אין שתי אבנים שעליהן אותו צירוף של מספרים. בסך הכול יש במחלקה **Stones** 49 אבני **BiStone**.

ד. כתוב את כותרת המחלקה **Stones** ואת התכונות שלה.

ה. כתוב במחלקה **Stones**, פעולה בונה שתיצור את כל 49 אבני ה-**BiStone**.

3.

[בגרות תשעז] תמונה נראית על מסך המחשב מורכבת מנקודות צבעוניות – פיקסלים. כל פיקסל מורכב משלושה רכיבי צבע: אדום, ירוק, כחול (Red Green Blue - RGB). כל רכיב צבע מיוצג על ידי מספר שלם בין 0 ל-255, והשילוב של שלושת הרכיבים קובע את צבע הפיקסל. לדוגמה: צבע לבן מיוצג על ידי המספרים (255,255,255), צבע כחול מיוצג על ידי המספרים (0,0,255), צבע אדום – (255,0,0), צבע שחור – (0,0,0).

א. כתוב את הכותרות ואת התכונות של המחלקה **Pixel** שמייצגת פיקסל.

לפניך תרשים חלקי של ממשק המחלקה **Pixel**:

<code>public Pixel()</code>	פעולה בונה היוצרת את הצבע השחור.
<code>public Pixel(int red, int green, int blue)</code>	פעולה בונה המקבלת 3 מספרים שלמים שהם הערכים של שלושת רכיבי הצבע של הפיקסל. אם אחד (לפחות) מן הפרמטרים לא נמצא בתחום הערכים המותר (0-255 [כולל]) – ייבנה עצם שצבעו שחור.
<code>public void Mix(Pixel pxl)</code>	הפעולה מקבלת עצם מטיפוס <code>Pixel</code> , ומשנה את ערכי הצבעים של העצם הנוכחי לצבע המתקבל מערבוב של שני צבעים (הצבע של העצם הנוכחי והצבע של העצם שהתקבל כפרמטר). ערבוב הצבעים המתקבל הוא ממוצע של שני הרכיבים.

שים לב: כאשר מתקבל ממוצע שהוא שבר (למשל 113.5) נלקח רק החלק השלם של הערך החדש (במקרה זה – 113).

לדוגמה: אם הצבע של העצם הנוכחי הוא אדום (255,0,0) והצבע של העצם שהתקבל כפרמטר הוא ירוק (0,255,0), אז הצבע המתקבל מן הערבוב הוא צהוב (127,127,0), והוא יהיה מעתה הבצע של העצם הנוכחי.

ב. ממש את הפעולות המופיעות בממשק המחלקה **Pixel**.

ג. הוסף מחלקה בשם **Pic50** שמייצגת תמונה בגודל 50 X 50.

1) למחלקה התכונה `pic` – מערך דו-ממדי בגודל 50 X 50 מטיפוס `Pixel`. כתוב את כותרת המחלקה ואת התכונה שלה.

2) כתוב במחלקה **Pic50** פעולה `Mix`, שתקבל תמונה. הפעולה תערבב את הצבעים של התמונה שהתקבלה עם הצבעים של התמונה הנוכחית ותעדכן את התמונה הנוכחית. הערבוב ייעשה בין פיקסלים שהאינדקסים שלהם זהים.

## רקורסיה

4. [בגרות תשס"ו] לפניך הפעולות Sod1, Sod2.

```

public static int Sod1(int n, int d, int a, Stack<int> S)
{
    S.Push(a);
    if (n == 1)
        return Sod2(S);
    else
        return Sod1(n - 1, d, d + a, S);
}

public static int Sod2(Stack<int> S)
{
    if (!S.IsEmpty())
    {
        int a = S.Pop();
        if (S.IsEmpty())
            return a;
        else
            return Sod2(S) + a;
    }
    else
        return 0;
}

```

א. לפניך המחסנית S1:

7
4
-5
2

עקוב אחר הזימון הבא: Sod2(S1), ורשום מה יוחזר.

ב. מה מבצעת הפעולה Sod2(S) עבור מחנית S כלשהי?

ג. בעבור מחסנית S2 ריקה, מה יחזיר הזימון Sod1(4, 6, 2, S2)? רשום את המעקב אחר ביצוע הפעולה Sod1. אין צורך להראות מעקב אחר ביצוע Sod2.

ד. מה מבצעת הפעולה Sod1(n, d, a, S) עבור מחסנית S ריקה, ובעבור מספרים שלמים a ו-d ומספר שלם n גדול מ-0?

5. [בגרות תשס"ז] נתונות הפעולות הבאות: Sod1, Sod2

```

public static int Sod1(int n1, int n2)
{
    if (n1 == 0 && n2 == 0)
        return 0;
    if (n1 == 0 && n2 != 0)
        return 1 + Sod1(0, n2 / 10);
    if (n1 != 0 && n2 == 0)
        return 1 + Sod1(n1 / 10, 0);
    return Sod1(n1 / 10, n2 / 10);
}

public static int Sod2(int k, int n, int[] a)
{
    if (k == n - 1)
        return Sod1(a[k + 1], a[k]);
    return Math.Max(Sod2(k + 1, n, a), Sod1(a[k + 1], a[k]));
}

```

א. מה יחזיר הזימון Sod1(86, 31547)? רשום את המעקב אחר הפעולה.

ב. מה מבצעת הפעולה Sod1(n1, n2) בעבור n1, n2 מספרים שלמים וגדולים מ-0?

ג. נתון מערך חד-ממדי a בגודל 5 :

a	9321	345	296	7	98
---	------	-----	-----	---	----

מה יחזיר הזימון Sod2(1, 5, a)? רשום את המעקב אחר ביצוע Sod2. אין צורך להראות מעקב אחר Sod1.

ד. מה מבצעת הפעולה Sod2 בעבור מערך a כלשהו בגודל n המכיל מספרים שלמים גדולים מ-0 ובעבור k שערכו גדול מ-0 וקטן מ-n?

6. [בגרות תשסח] נתונות הפעולות הבאות : Sod1, Sod2, Sod3, Sod4.

```
public static int Sod1(int p, int p1, int n, int[] a)
```

```
{
    if (p == p1)
        return 0;
    int x = Math.Abs(a[p] - a[p1]);
    int y = Sod1(p, p1 + 1, n, a);
    return Math.Max(x, y);
}
```

```
public static int Sod2(int p, int p2, int n, int[] a)
```

```
{
    if (p == p2)
        return 0;
    int x = Math.Abs(a[p] - a[p2]);
    int y = Sod2(p2 - 1, p, n, a);
    return Math.Max(x, y);
}
```

```
public static int Sod3(int n, int p, int[] a)
```

```
{
    return Math.Max(Sod1(p, 1, n, a), Sod2(n, p, n, a));
}
```

```
public static int Sod4(int n, int[] a)
```

```
{
    int k = Sod3(n, 1, a);
    for (int i = 2; i <= n; i++)
    {
        int t = Sod3(n, i, a);
        if (k < t)
            k = t;
    }
    return k;
}
```

נתון מערך חד-ממדי a בגודל 5 (משמאל לימין) :

a	4	3	7	6	10
---	---	---	---	---	----

א. מה יחזיר הזימון Sod3(5, 4, a)? רשום את המעקב.

ב. מה מבצעת הפעולה Sod3(n, p, a) עבור מערך a בגודל n ומספר שלם p, שערכו גדול מ-0 וקטן או שווה ל-n?

ג. מה יחזיר הזימון Sod4(5, a)? רשום את המעקב אחר ביצוע Sod4. אין צורך להראות את המעקב אחר ביצוע Sod3.

ד. מה מבצעת הפעולה Sod4(n, a) בעבור מערך a בגודל n?

## 7. [בגרות תשס"ט] לפניך שתי פעולות חיצוניות:

```

public static int Sod1(Queue<int> q)
{
    int i = q.Remove();
    int result = i;
    if (!q.IsEmpty())
    {
        int j = Sod1(q);
        if (result > j)
            result = j;
    }
    q.Insert(i);
    return result;
}

public static int Sod2(int i)
{
    if (i == 0)
        return 0;
    int a = i % 10;
    int b = Sod2(i / 10);
    if (a > b)
        return a;
    return b;
}

```

נתון התור myQueue מטיפוס Queue<int>

הוצאת ערכים => 

2	31	4	17	5
---	----	---	----	---

 => הכנסת ערכים

א. מה יחזיר הזימון Sod1(myQueue)? רשום את המעקב. רשום את התור המתקבל בתום המעקב. ציין את ראש תור ואת סוף התור.

ב. מה מבצעת הפעולה Sod1(queue) בעבור תור queue לא ריק מטיפוס Queue<int>?

ג. מה יחזיר הזימון Sod2(17852)? רשום את המעקב.

ד. מה מבצעת הפעולה Sod2(k) בעבור מספר k גדול מ-0 וטיפוס שלם?

ה. מה מבצעת הפעולה Sod2(Sod1(queue)) בעבור תור queue לא ריק מטיפוס Queue<int> המכיל רק מספרים שלמים וגדולים מ-0?

8. [בגרות תשע] לפניך שתי פעולות חיזוניות:

```
public static void Sod1(Queue<int> qu, Stack<int> st)
{
    if (!qu.IsEmpty())
    {
        int x = qu.Remove();
        Sod1(qu, st);
        st.Push(x);
        qu.Insert(x);
    }
}

public static void Sod2(Queue<int> qu, Stack<int> st)
{
    if (!qu.IsEmpty())
    {
        int x = qu.Remove();
        Sod2(qu, st);
        int y = st.Pop();
        qu.Insert(x + y);
    }
}
```

נתון התור myQueue מטיפוס Queue<int> הוצאת ערכים => 

9	8	1	3	4
---	---	---	---	---

 הכנסת ערכים => ונתון קטע התכנית הבא:

```
Stack<int> myStack = new Stack<int>();
Sod1(myQueue, myStack);
Sod2(myQueue, myStack);
```

א. עקוב אחר הביצוע של קטע התכנית ורשום את המעקב.

ב. ענה על תתי-הסעיפים הבאים על פי המעקב שרשמת:

1) כתוב מה הכיל התור myQueue ומה הכילה המחסנית myStack לאחר הזימון:

```
Sod1(myQueue, myStack);
```

2) כתוב מה מכיל התור myQueue ומה מכילה המחסנית myStack בסוף קטע התכנית.

ג. מה מבצעת הפעולה Sod1 בעבור תור כלשהו לא ריק myQueue מטיפוס Queue<int> ומחסנית myStack ריקה מטיפוס Stack<int>?

ד. מה מבצע קטע התכנית בעבור תור כלשהו לא ריק myQueue מטיפוס Queue<int> ומחסנית myStack ריקה מטיפוס Stack<int>?

## 9. [בגרות תשעא] לפניך כמה מחלקות.

א. עקוב אחר הפעולה Main שבמחלקת Program ורשום את הפלט. במעקב יש לכלול את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

```
public class One
{
    private double n;
    public One(double n) { this.n = n; }
    public double GetN() { return this.n; }
    public void F() { Console.WriteLine("F of one"); }
    public void G() { Console.WriteLine("G of one " + this.GetN()); }
}

public class Two
{
    private Queue<One> q;
    public Two(int n, int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.Insert(new One(Math.Pow(2, i)));
    }
    public void F()
    {
        Console.WriteLine("F of two");
        if (!this.q.IsEmpty())
            Console.WriteLine(this.q.Remove().GetN());
    }
}

public class Together
{
    private double x;
    private int from, to;
    private One first;
    private Two second;
    public Together(double x, int from, int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from, to);
    }
    public void MethodA()
    {
        Console.WriteLine("-- MethodA() --");
        this.first.F();
    }
    public void MethodB()
    {
        Console.WriteLine("-- MethodB() --");
        this.first.G();
        this.second.F();
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Together tg = new Together(5.0, 2, 6);
        tg.MethodA();
        tg.MethodB();
    }
}
```

ב. בהנחה כי הפעולה הבונה במחלקה Together מקבלת מספרים גדולים מ-0 בלבד – מה מבצעת הפעולה F() במחלקה Two?



10. [בגרות תשעא] לפניך הפעולות Sod ו-What המקבלות מערך  $a$  שאיבריו מטיפוס שלם, ממין בסדר עולה, ומספר שלם. לשתי הפעולות אותה טענת יציאה.

```
public static bool Sod(int[] a, int k)
{
    for (int i = 0; i < a.Length - 1; i++)
    {
        int j = i + 1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static bool What(int[] a, int k)
{
    int left = 0, right = a.Length - 1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

נתון מערך  $a$ :

$a$	2	4	7	12	18
-----	---	---	---	----	----

- א. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון  $a$  והמספר  $k=11$ , ורשום את הערך המוחזר. בטבלת המעקב יש לכלול עמודות בעבור:  $i, j, a[i], a[j]$  ועמודה נוספת שבה יצוין אם תנאי שבפקודת if מתקיים או אינו מתקיים.
- ב. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון  $a$  והמספר  $k=10$ . רשום את הערך המוחזר. בטבלת המעקב יש לכלול את העמודות שפורטו בסעיף א.
- ג. מהי טענת היציאה של הפעולה Sod?
- ד. מהי סיבוכיות זמן הריצה של הפעולה Sod? נמק את תשובתך.
- ה. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון  $a$  והמספר  $k=11$ , ורשום את הערך המוחזר. בטבלת המעקב יש לכלול עמודות בעבור:  $left, right, a[left], a[right]$  ושתי עמודות נוספות לכל אחת מפקודות if. בכל עמודה יצוין אם התנאי בפקודת if מתקיים או אינו מתקיים.
- ו. מהי סיבוכיות זמן הריצה של הפעולה What? נמק את תשובתך.
- ז. מי מבין שתי הפעולות – Sod או What – יעילה יותר? נמק את תשובתך.
- ח. טענת הכניסה של הפעולות Sod ו-What שונתה כך שאפשר להעביר אליהן מערך  $a$  לא ממין.
  - 1) האם טענת היציאה של הפעולה Sod תשתנה? נמק את תשובתך.
  - 2) האם טענת היציאה של הפעולה What תשתנה? נמק את תשובתך.

11. [בגרות תשעב] לפניך הפעולה Karamnba המקבלת מספר שלם  $k$ , מערך  $a$  שאיבריו מטיפוס שלם, ושני מספרים שלמים נוספים  $s, e$  כאשר  $s \leq e$ .

```
public static int Karamba(int k, int[] a, int s, int e)
{
    if (s == e)
        if (a[s] > k)
            return a[s];
        else
            return 0;
    else
    {
        int p1 = Karamba(k, a, s, (s + e) / 2);
        int p2 = Karamba(k, a, ((s + e) / 2) + 1, e);
        return p1 + p2;
    }
}
```

א. נתון מערך  $a$ :

A	2	8	4	14	5	18
---	---	---	---	----	---	----

לפניך זימון הפעולה Karamba: `Console.WriteLine(Karamaba(6, a, 1, 5));`

עקוב אחר ביצוע הפעולה בעבור המערך  $a$  והזימון הנתון, ורשום מה יהיה הפלט כתוצאה מזימון הפעולה.

ב. בעבור המערך  $b$  לפניך:

B	2	2	2	2	2	2
---	---	---	---	---	---	---

תן דוגמה לזימון הפעולה Karamba שבעבורו יוחזר הערך 6.

ג. מה מבצעת הפעולה Karamba?

ד. מבי סיבוכיות זמן הריצה של הפעולה Karamba? נמק את תשובתך.

12. [בגרות תשעג] לפניך שלוש פעולות Sod1, Sod2, Sod3.

```
public static bool Sod1(Node<int> list1, Node<int> list2)
{
    Node<int> node1 = list1;
    Node<int> node2 = list2;
    for (int i = 1; i <= 4; i++)
    {
        if ((node1 == null) || (node2 == null))
            return false;
        if ((i == 1) || (i == 4))
            if (node1.GetValue() != node2.GetValue())
                return false;
        node1 = node1.GetNext();
    }
    return true;
}

public static bool Sod2(Node<int> list1, Node<int> list2)
{
    Node<int> node1 = list1;
    Node<int> node2 = list2;
    while ((node1 != null) && (node2 != null))
    {
        if (node1.GetValue() != node2.GetValue())
            return false;
        node1 = node1.GetNext();
        node2 = node2.GetNext();
    }
    return true;
}
```

```

public static bool Sod3(Node<int> list1, Node<int> list2)
{
    Node<int> node1 = list1;
    while (node1 != null)
    {
        bool found = false;
        Node<int> node2 = list2;
        while ((node2 != null) && (!found))
        {
            if (node1.GetValue() == node2.GetValue())
                found = true;
            node2 = node2.GetNext();
        }
        if (!found)
            return false;
        node1 = node1.GetNext();
    }
    return true;
}

```

א. עקוב אחר כל אחת מהפעולות Sod1, Sod2, Sod3, עם הרשימות list1 ו-list2 שלפניך:

list1: 2 → 4 → 5 → 1 → 1 → 9 → ||

list2: 2 → 4 → 5 → 1 → 4 → ||

רשום מה יוחזר בעבור כל אחת מהפעולות. במעקב הראה את המעבר על הרשימות.

ב. מה סיבוכיות הריצה של כל אחת מהפעולות Sod1, Sod2, Sod3? נמק את תשובתך.

13. [בגרות תשעד] נתונה פעולה המקבלת רשימה lst של מספרים שלמים:

```

public static int What(Node<int> lst)
{
    if (lst == null)
        return 0;
    int x = lst.GetValue();
    lst = lst.GetNext();
    if (x >= 0)
        return What(lst); // (*)
    return 1 + What(lst);
}

```

נתונה הרשימה:

lst: -2 → -9087 → 16 → -43 → 5 → ||

א. עקוב אחר הפעולה הנתונה בעבור הרשימה lst. רשום את הערך שיוחזר, וסרטט את הרשימה לאחר ביצוע הפעולה.

ב. מה הפעולה מבצעת?

ג. החליפו את ההוראה המסומנת ב-(\*) להוראה: return 0;

(1) מה תבצע הפעולה לאחר ההחלפה?

(2) מה תכיל הרשימה lst לאחר ביצוע הפעולה?

14. [בגרות תשעד] לפניך הפעולה Sum הכתובה במחלקה ראשית.

```
public static int Sum(Stack<int> s)
{
    if (s.IsEmpty())
        return 0;
    int x = s.Pop();
    if (x % 6 == 0)
        return x + Sum(s);
    return Sum(s);
}
```

נתונה המחסנית s.

ראש המחסנית	12
	4
	33
	6
	30
	0

א. עקוב אחר הפעולה Sum בעבור המחסנית s, ורשום את הערך שיוחזר. במעקב הראה את המעבר על המחסנית s.

ב. לפניך הפעולה Sod.

```
public static void Sod(Queue<Stack<int>> qq, Queue<int> qm)
{
    if (!qq.IsEmpty())
    {
        int x = Sum(qq.Remove());
        qm.Insert(x);
        Sod(qq, qm);
    }
}
```

ג. לפניך קטע תכנית המשתמש בפעולה Sod. סרטט במחברתך את כל אחד משני התורים q1 ו-q2, לפני הקריאה לפעולה Sod ואחר ביצוע הפעולה Sod.

```
public static void Main(string[] args)
{
    Stack<int> s1 = new Stack<int>();
    Stack<int> s2 = new Stack<int>();

    Queue<Stack<int>> q1 = new Queue<Stack<int>>();
    s1.Push(0);
    s1.Push(30);
    s1.Push(6);
    s1.Push(33);
    s1.Push(4);
    s1.Push(12);
    s2.Push(23);
    s2.Push(36);
    s2.Push(1);
    q1.Insert(s1);
    q1.Insert(s2);
    Queue<int> qr = new Queue<int>();
    Sod(q1, qr);
}
```

15. [בגרות תשעה] לפניך קטע תכנית. נתון ש- $a$  הוא ערך מטיפוס שלם בגודל  $n$ .

```
int i = 0;
while (i < n - 1)
{
    if (a[i] > a[i + 1])
    {
        a[i] = a[i] + a[i + 1];
        a[i + 1] = a[i] - a[i + 1];
        a[i] = a[i] - a[i + 1];
        i = 0;
    }
    else
        i++;
}
```

א. עקוב אחר קטע התכנית בעבור המערך  $a$  בגודל 4 שלפניך.

a	5	7	8	12
---	---	---	---	----

במעקב יש להראות את  $i$  ואת המערך בסיום כל איטרציה.

ב. מהי סיבוכיות זמן הריצה של קטע התכנית בעבור מערך בגודל  $n$  ממיון בסדר עולה? נמק את תשובתך.

ג. עקוב אחר קטע התכנית בעבור המערך  $a$  בגודל 4 שלפניך.

a	12	8	7	5
---	----	---	---	---

במעקב יש להראות את  $i$  ואת המערך בסיום כל איטרציה.

ד. מהי סיבוכיות זמן הריצה של קטע התכנית בעבור מערך בגודל  $n$  ממיון בסדר יורד? נמק את תשובתך.

16. [בגרות תשעו] לפניך פעולה המקבלת כפרמטר תור  $q$  של מספרים שלמים הממוינים בסדר עולה. כל המספרים שונים זה מזה. הנח שבתור יש לפחות שני מספרים.

```
public static int What(Queue<int> q)
{
    int x = q.Remove();
    int y = q.Head();
    q.Insert(x);
    if (x > y)
        return 0;
    return (What(q) + (y - x));
}
```

א. נתון התור  $q_1$ .

הכנסת ערכים	20	17	8	4	1	הוצאת ערכים
-------------	----	----	---	---	---	-------------

עקוב אחר הפעולה  $What$  בעבור התור הנתון  $q_1$ , וכתוב את הערך שיוחזר. הראה את המעקב הרקורסיבי. במעקב הראה את הערכים של  $x$  ושל  $y$  ואת התור לאחר כל זימון רקורסיבי.

ב. מה מחזירה הפעולה  $What$  בעבור תור כלשהו  $q$  של מספרים שלמים, הממוינים בסדר עולה ושונים זה מזה?

ג. כיצד נראה התור לאחר סיום הפעולה?

ד. אם נחליף את התנאי  $if(x>y)$  בתנאי  $if(q.IsEmpty())$  תיווצר שגיאת זמן ריצה. מה השגיאה שתיווצר? הסבר את תשובתך.

17. [בגרות תשעו] לפניך פעולה המקבלת מערך חד-ממדי ar מטיפוס שלם, ושני מספרים x ו-y גדולים מ-0 או שווים לו, וקטנים מגודל המערך, ו- $x \leq y$ .

```
public static int Go(int[] ar, int x, int y)
{
    if (x == y) return (ar[x] % 10);
    if ((ar[x] % 10) < (ar[y] % 10))
        return Go(ar, x + 1, y);
    return Go(ar, x, y - 1);
}
```

לפניך מערך ar1 מטיפוס שלם.

ar1	21	219	166	61	127	60	34
-----	----	-----	-----	----	-----	----	----

- א. מה יחזיר הזימון `Go(ar1, 2, 5)`? הראה את המעקב הרקורסיבי. במעקב הראה את הערכים של: `ar1[y]`, `ar1[x]`, `y`, `x`
- ב. מה יחזיר הזימון `Go(ar1, 0, ar1.Length-1)`?
- ג. מה מבצעת הפעולה `public static int Go(int[] ar, int x, int y)` בעבור מערך ar כלשהו מטיפוס שלם ושני מספרים שלמים x ו-y גדולים מ-0 או שווים לו, וקטנים מגודל המערך, ו- $x \leq y$ .
18. [בגרות תשעז] לפניך תכנית. הפעולה What מקבלת מערך חד-ממדי המכיל מספרים שלמים, ומספר שלם k,  $k > 0$ , כל המספרים במערך הם בין 0 ל-k (כולל).

```
public class Program
{
    public static int[] What(int[] arr, int k)
    {
        int n = arr.Length;
        int[] b = new int[n];
        int[] c = new int[k + 1];
        for (int i = 0; i < k + 1; i++) c[i] = 0;
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1; /**

        for (int i = 1; i < k + 1; i++) c[i] = c[i] + c[i - 1]; /**

        for (int j = 0; j >= 0; j--)
        {
            b[c[arr[j]] - 1] = arr[j];
            c[arr[j]] = c[arr[j]] - 1;
        }
        return b;
    }

    public static void Main(string[] args)
    {
        int[] arr = { 5, 0, 2, 1, 3, 5 };
        arr = What(arr, 5);
        for (int i = 0; i < arr.Length; i++)
            Console.WriteLine(arr[i]);
    }
}
```

- א. סרטט את המערך c לאחר ביצוע הלולאה המסומנת `/**`.
- ב. סרטט כיצד יראה המערך לאחר ביצוע הלולאה המסומנת `/**`.
- ג. עקוב אחר ביצוע הלולאה המסומנת `/**`. במעקב עליך להראות את: `b[c[arr[j]]-1]`, `c[arr[j]]`, `j`, `arr[j]`, המערך b והמערך c.
- ד. מה מבצעת הפעולה `What`?
- ה. מהי סיבוכיות זמן הריצה של הפעולה `What`? נמק את תשובתך.

## רשימה

**19.** [בגרות תשסו] כתוב פעולה שתקבל שתי רשימות  $L1$ ,  $L2$  של מספרים שלמים וגדולים מ-0, ותחזיר רשימה חדשה  $L3$ . הפעולה תסרוק את הרשימה  $L1$  פעם אחת, מתחילתה ועד סופה. בכל שלב הפעולה תבדוק איבר אחד מהרשימה  $L1$ . נסמן את ערך האיבר ב- $k$ . הפעולה תבצע את אחת הפעולות שלהלן, בהתאם ל- $k$ , ולרשימה  $L2$  כפי שהיא בשלב זה:

- אם  $k$  הוא מספר זוגי, הפעולה תמחק את האיבר שמקומו  $k$  מתחילת הרשימה  $L2$ .
- אם  $k$  הוא מספר אי-זוגי, הפעולה תוסיף לרשימה  $L3$  איבר שהערך שלו הוא הערך של האיבר שמיקומו  $k$  מתחילת הרשימה  $L2$ . אך אם ברשימה  $L2$  אין איבר שמיקומו  $k$  התוכנית לא יתבצע דבר.

הערות: אין חשיבות לסדר הכנסת האיברים לרשימה  $L3$ , והפעולה לא תחזיר את הרשימה  $L2$ .  
לדוגמא:

בעבור הרשימות  $L1$ ,  $L2$  (משמאל לימין):

$L1$ : 4 , 3 , 2 , 6  
 $L2$ : 10 , 11 , 19 , 1 , 7 , 100

מצב הרשימות  $L2$ ,  $L3$  בתום כל שלב של ביצוע הפעולה יהיה כזה:

○ אחרי שלב א

$L2$ : 10 , 11 , 19 , 7 , 100  
 $L3$ :

○ אחרי שלב ב

$L2$ : 10 , 11 , 19 , 7 , 100  
 $L3$ : 19

○ אחרי שלב ג

$L2$ : 10 , 19 , 7 , 100  
 $L3$ : 19

○ אחרי שלב ד

$L2$ : 10 , 19 , 7 , 100  
 $L3$ : 19

**20.** [בגרות תשסז] בטלפון נייד של חברת "נייד-פון" ניהול השיחות הנכנסות מתבצע באופן הזה:

- מספר טלפון שממנו התקבלה שיחה נשמר ברשימת שיחות נכנסות.
- מספרי הטלפון של השיחות הנכנסות שהתקבלו במכשיר נשמרים לפי סדר הפוך מסדר קבלתן, כך שמספר הטלפון של השיחה האחרונה שהתקבלה יהיה ראשון ברשימת השיחות הנכנסות.
- כל מספרי הטלפון השמורים שונים זה מזה. כאשר מתקבלת שיחה נוספת ממספר טלפון הנמצא כבר בין מספרי השיחות הנכנסות מספר זה נרשם ראשון והוא נמחק ממקומו הקודם.
- מספר השיחות הנכנסות שאפשר לשמור במכשיר מוגבל.
- כאשר רשימת השיחות הנכנסות מלאה ומתקבלת שיחה ממספר שטלפון שאינו נמצא ברשימה, מספר זה נרשם ראשון והמספר האחרון נמחק.

**יומן שיחות נכנסות** מיוצג על ידי :

- i. רשימה שכל איבר בה הוא מספר טלפון המיוצג על ידי מחרוזת.
- ii. מספר שלם maxCalls המייצג את המספר המקסימלי של מספרי הטלפון הנמצאים בזמן מסוים ביומן השיחות הנכנסות.
- iii. מספר שלם currentCalls המייצג את המספר של המספרי הטלפון הנמצאים בזמן מסוים ביומן השיחות הנכנסות.

א. כתוב את כותרת המחלקה **יומן שיחות נכנסות**, על פי הייצוג המתואר ב- i-iii.

ב. ממש פעולה במחלקה **יומן שיחות נכנסות** המקבלת מספר טלפון של שיחה נכנסת tel ומעדכנת את יומן השיחות הנכנסות.

**21.** [בגרות תשסח] טיפוס הנתונים **מיון-תחומים** מאפשר לשמור כמות גדולה של מספרים שלמים, שונים זה מזה, הממוינים לפי תחומים. לכל תחום מוגדר ערך מינימלי וערך מקסימלי. כל תחום כולל מספרים הגדולים מהערך המינימלי או שווים לו, וקטנים מהערך המקסימלי. לתחום האחרון מוגדר רק ערך מינימלי, והוא כולל מספרים הגדולים מערך זה או שווים לו.

טיפוס הנתונים **מיון-תחומים** מורכב מ :

- mins : מערך חד-ממדי בגודל 100 המכיל מספרים שלמים שונים זה מזה, הממוינים בסדר עולה. כל מספר מייצג ערך מינימלי של התחום.

- values : מערך חד-ממדי בגודל 100 המכיל רשימות. כל רשימה מכילה מספרים שלמים שונים זה מזה, הממוינים בסדר עולה. ברשימה שנמצאת במקום ה-k במערך values נמצאים המספרים גדולים מן הערך שנמצא במקום ה-k במערך mins או שווים לו, וקטנים מן הערך במקום ה-k+1 במערך mins. ברשימה האחרונה נמצאים מספרים הגדולים מן הערך המינימלי של התחום האחרון או שווים לו.

- current : מספר שלם המייצג את מספר התחומים הנוכחי שיש במערך mins.

לדוגמה : בעבור המספרים האלה : 1, 700, 556, 120, 220, 9, 6, 99, 555 :

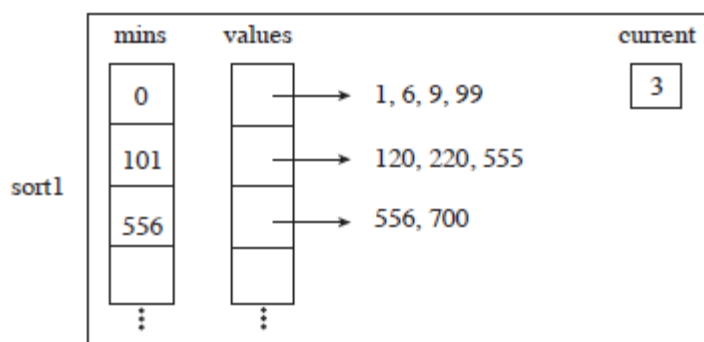
ושלושת התחומים האלה :

i. מספרים הגדולים מ-0 או שווים לו וקטנים מ-101.

ii. מספרים הגדולים מ-101 או שווים לו וקטנים מ-556.

iii. מספרים השווים ל-556 או גדולים ממנו.

Sort1 מטיפוס **מיון-תחומים** יהיה :



א. רשום את כותרת המחלקה **מיון-תחומים SortByRange** ואת התכונות שלה.

ב. ממש פעולה פנימית בשם Insert במחלקה **SortByRange** שתקבל מספר שלם num. הפעולה תכניס את num לרשימה המתאימה במערך values על פי סדר המיון של המספרים ברשימה. הנח כי num אינו נמצא ב-values ו-num גדול מהערך הקטן ביותר שנמצא במערך mins.

ג. מהי סיבוכיות זמן הריצה של הפעולה שמימשת בסעיף ב? נמק.



ד. ממש פעולה פנימית בשם `AddRange` במחלקה `SortByRange`, שתקבל מספר שלם `r` הגדול מכל המספרים ב-`values`. הפעולה תוסיף תחום חדש ש-`r` הוא הערך המינימלי שלו. הנח כי יש מקום לתחום נוסף.

ה. מהי סיבוכיות זמן הריצה של הפעולה שמימשת בסעיף ד? נמק.

22. [בגרות תשסט] רשימה `L` תקרא רשימה **משולשת** אם היא מקיימת את התנאים הבאים:

- הרשימה אינה ריקה.
- מספר האיברים בה מתחלק ב-3 ללא שארית.
- האיברים בשליש הראשון של הרשימה מכילים את אותם ערכים שמכילים האיברים בשליש השני של הרשימה ואתם ערכים שמכילים האיברים בשליש השלישי של הרשימה. הערכים מסודרים באותו הסדר בכל אחד מהשלישים.

לדוגמה הרשימה `L1` שלפניך היא רשימה משולשת באורך 12.

`L1: 2 → 5 → 3 → 7 → 2 → 5 → 3 → 7 → 2 → 5 → 3 → 7 → null`

כתוב פעולה חיצונית שתקבל רשימה `L` האיברים שלה הם מטיפוס מספר שלם.

אם `L` היא רשימה **משולשת**, הפעולה תחזיר `true`, אחרת – הפעולה תחזיר `false`.

הארה: אתה יכול להשתמש בפעולות של מחלקת `Node<T>` מבלי לממש אותן. אם אתה משתמש בפעולות אחרות עליך לממשן. אין צורך לשמור על שלימות הרשימה `L`.

23. [בגרות תשע] `L` היא רשימה המכילה מספרים שלמים שונים זה מזה וממוינים בסדר עולה. **רשימת הטווחים** של `L` היא רשימה חדשה שנבנית באופן הזה: בעבור כל רצף של מספרים עוקבים ב-`L` יהיה

ברשימת הטווחים איבר אחד המכיל שני מספרים, מספר אחד הוא המספר הקטן ביותר ברצף, והמספר השני הוא המספר הגדול ביותר ברצף. רצף יכול להיות באורך 1 או יותר. אם הרצף הוא באורך 1, הוא יוצע ברשימת הטווחים על ידי איבר ששני המספרים בו שווים.

לדוגמה, בעבור הרשימה `L` שלפניך:

`3 → 4 → 5 → 12 → 19 → 20 → 100 → 101 → 102 → 103 → 104 → null`

רשימת הטווחים של `L` תהיה:

`3 5 → 12 12 → 19 20 → 100 104 → null`

לפניך תיאור חלקי של המחלקה `RangeNode`, המייצגת איבר ברשימת הטווחים.

RangeNode	
<code>private int from;</code>	// המספר הקטן ביותר ברצף
<code>private int to;</code>	// המספר הגדול ביותר ברצף
<code>public RangeNode(int from, int to)</code>	

ממש פעולה חיצונית שתקבל רשימה לא ריקה, המכילה מספרים שלמים שונים זה מזה וממוינים בסדר עולה, ותחזיר את רשימת הטווחים שלה.

כותרת הפעולה: `public static Node<RangeNode> CreateRangeList(Node<int> sourceList)`

הנח שלכל אחת מהתכונות במחלקה `RangeNode` יש פעולות `Get` ו-`Set`. אתה יכול להשתמש בפעולות אלו ובפעולה הבונה של המחלקה `RangeNode` מבלי לממש אותן.

## 24. [בגרות תשעג]

- א. כתוב פעולה שתקבל מספר שלם num גדול מ-1, ורשימה lst המכילה מספרים שלמים גדולים מ-0, שכולם קטנים מ-num. הפעולה תחזיר רשימה חדשה שאיבריה הם כל המספרים השלמים הגדולים מ-0 וקטנים מ-num שאינם מופיעים ברשימה lst.
- ב. מהי סיבוכיות זמן הריצה של הפעולה שכתבת? נמק את תשובתך.

25. [בגרות תשעד] לפניך שלוש מחלקות: המחלקה **Birth** – תאריך לידה, שתכונותיה הן: יום בחודש – day,

חודש - month, ושנה – year. כל אחת מן התכונות היא מספר שלם.

המחלקה **Student** – תלמיד, שתכונותיה הן: שם התלמיד – name מטיפוס מחרוזת, ותאריך לידה של התלמיד birthday מטיפוס **Birth**.

המחלקה **School** – בית ספר שתכונתה היא: מערך חד-ממדי ar בגודל 6, מטיפוס `Node<Student>`. כל תא במערך ar מייצג שכבת גיל בבית הספר: תא 0 מייצג שכבה ז', תא 1 מייצג שכבה ח' וכך הלאה, עד שתא 5 שמייצג את שכבה י"ב. כל תא מכיל רשימה של כל תלמידי השכבה.

הנח שבכל מחלקה יש: פעולה בונה בררת מחדל ופעולה בונה שמקבלת פרמטר לכל תכונה, פעולות Get ו-Set לכל תכונה.

כתוב פעולה חיצונית שתקבל עצם מטיפוס **School** ותחזיר מערך חד-ממדי בגודל 12 מטיפוס `Node<Student>`. כל תא במערך מייצג חודש בשנה: 0 מייצג את ינואר, תא 1 מייצג את פברואר וכך הלאה, עד תא 11 שמייצג את דצמבר. כל תא יכיל רשימה של כל התלמידים מכל השכבות שנולדו בחודש זה, בלי חשיבות לסדר.

26.

[בגרות תשעה] המחלקה מסלול-אוטובוס – **BusRoute** מתארת מסלול של קו אוטובוס. כל מסלול מכיל מספר כלשהו של תחנות, לכל הפחות שתיים, בסדר מסוים. כל תחנה מיוצגת על ידי שני מספרים שלמים המציינים את מיקומה במישור. כל תחנה מופיעה במסלול פעם אחת. לדוגמה, מסלול בן חמש תחנות המתחיל בתחנה (0, 2) ומסתיים בתחנה (5, 0):

$(0, 2) \rightarrow (1, 4) \rightarrow (5, 4) \rightarrow (3, 1) \rightarrow (5, 0)$

האוטובוס נוסע מתחנה לתחנה, לפי סדר התחנות. אורך המסלול הוא סכום המרחקים בין תחנה לתחנה. הנח שנתונה המחלקה תחנה – **Station** שתכונותיה שני מספרים שלמים x ו-y המציינים את מיקומה של התחנה במישור. במחלקה **Station** הוגדרו שתי הפעולות:

פעולה בונה `Station (int x, int y)`

פעולה שכותרתה `double Distance(Station other)`

הפעולה מקבלת תחנה `other` ומחזירה את המרחק בין התחנה הנוכחית ובין התחנה `other`. לפניך ממשק המחלקה מסלול-אוטובוס **BusRoute**:

פעולה	תיאור
<code>BusRoute(Station first, Station second)</code>	פעולה בונה המקבלת שתי תחנות, ויוצרת מסלול-אוטובוס עם שתי התחנות.
<code>void AddStation(Station newStation)</code>	פעולה המקבלת תחנה נוספת ומוסיפה אותה בסוף מסלול אוטובוס הקיים. הנח שתחנה זו אינה קיימת במסלול.
<code>double RouteLength()</code>	פעולה המחזירה את אורך המסלול-אוטובוס, כלומר את סכום המרחקים בין תחנה לתחנה.

א. כתוב את כותרת המחלקה **BusRoute** ואת התכונה/התכונות שלה. כתוב תיעוד לכל תכונה.

ב. ממש את הפעולה הבונה של המחלקה **BusRoute**.

ג. ממש את הפעולה המוסיפה תחנה למסלול-אוטובוס.

ד. ממש את הפעולה המוסיפה תחנה למסלול-אוטובוס.

ה. כתוב בפעולה הראשית במחלקה `Program` קטע תכנית שייצור את המסלול-אוטובוס שבדוגמה שבתחילת השאלה, וידפיס את אורך המסלול-אוטובוס.

הערה: אתה יכול להשתמש בפעולות המחלקה **Station** בלי לממש אותן.

27. [בגרות תשעו] נגדיר **רשימה דו-כיוונית** כאוסף סדוק של חוליות מטיפוס `BinNode<int>` המקושרות כך :  
 לכל זוג חוליות `p1` , `p2` ברשימה, אם מתקיים `p1.GetRight() == p2` , אז מתקיים גם `p2.GetLeft() == p1`.  
**ברשימה דו-כיוונית** 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 שאחריה.  
 לפניך דוגמה ל**רשימה דו-כיוונית** ומשתנה `pos` מטיפוס `BinNode<int>` המצביע על חוליה כלשהי ברשימה דו-כיוונית.



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

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

א. לפניך שלד 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 מבציע לחוליה כלשהי **ברשימה דו-כיוונית** ומחזירה ערך בוליאני. **הרשימה הדו-כיוונית** מכילה לפחות 3 חוליות.

1) עקוב אחר ביצוע הפעולה בעבורה המשתנה `pos` והרשימה שבדוגמה המצוגת בתחילת השאלה. במעקב הראה את הרשימה הדו-כיוונית ואת ערכי המשתנים : `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 != null) && (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 בפעולה – השורות המוקפות

במסגרת – בהוראה : `return left.GetValue() + right.GetValue() == sum;` נמק את קביעתך.

## מחסנית ותור

28. [בגרות תשס"ו] לפניך פעולה.

Queue<int> QByFrequency (Queue<int> q)	<p>הפעולה מקבלת תור q המכיל מספרים שלמים, ומחזירה תור חדש. בעבור כל מספר בתור q יהיו בתור החדש שני איברים: האיבר הראשון מכיל את המספר מהתור q, והאיבר השני מכיל את מספר הפעמים שהוא מופיע בתור q. בעבור מספר המופיע יותר מפעם אחת בתור q יהיה זוג אחד בלבד בתור החדש.</p>
--	---

לדוגמה: נתון התור Q (משמאל לימין):

=> 

1	4	4	1	5	-9	1	-9	-9
---	---	---	---	---	----	---	----	----

 =>

התור שיוחזר לאחר זימון הפעולה יהיה (משמאל לימין):

=> 

1	3	4	2	5	1	-9	3
---	---	---	---	---	---	----	---

 =>

א. ממש את הפעולה QByFrequency.

ב. מהי סיבוכיות זמן הריצה של הפעולה שכתבת בסעיף א? נמק.

29. [בגרות תשס"ח] **איבר מחליף כיוון** הוא מספר בסדרת מספרים שבו משתנה סדר המיון בין איברים סמוכים

מעלייה לירידה או מירידה לעלייה. **איבר מחליף כיוון** שייך גם לרצף של מספרים הממוינים בסדר עולה וגם לרצף של מספרים הממוינים בסדר יורד.

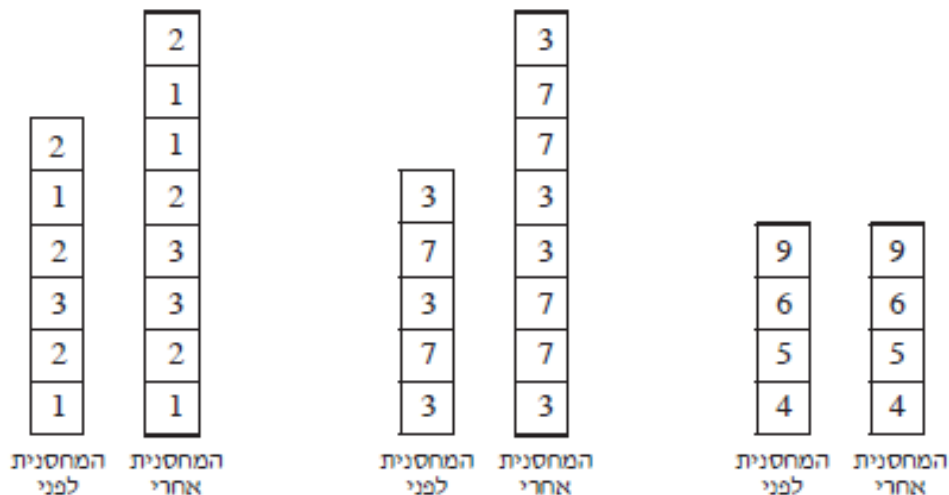
לדוגמה: בסדרה – 2, 5, 7, 4, 2, 1, 8

המספרים 7 ו-1 הם **איברים מחליפי כיוון**.

א. כתוב פעולה שתקבל מחסנית לא ריקה של מספרים שלמים, שאין בה מספרים זהים צמודים זה לזה.

הפעולה תכניס למחסנית אחרי כל איבר מחליף כיוון איבר זהה לו.

דוגמאות:



ב. מהי סיבוכיות זמן הריצה של הפעולה שכתבת בסעיף א? נמק.

30. [בגרות תשע] כתוב פעולה חיצונית המקבלת שתי מחסניות st1, st2 המכילות מספרים שלמים וגדולים מ-0. הפעולה תחזיר את הסכום של זוג האיברים הסמוכים הקרוב ביותר לראש המחסנית st1, שסכומם גדול מהסכום של כל זוג איברים סמוכים במחסנית st2. אם אין זוג כזה – הפעולה תחזיר 0. הנח שבכל אחת מהמחסניות יש לפחות שני איברים. לדוגמה:

2	
8	7
4	9
13	12
4	8
1	14
9	6
11	7
st2	st1

סכום האיברים הסמוכים 9-12 במחסנית st1 גדול יותר מסכום כל שני איברים סמוכים במחסנית st2, ולכן הפעולה תחזיר 21. שים לב: במחסנית st1 יש גם זוג איברים סמוכים שסכומם גדול יותר מ-21, אך הוא מופיע עמוק יותר במחסנית מזוג האיברים הסמוכים 9-12. הערה: אין צורך לשמור על תוכן המקורי של המחסניות.

31. [בגרות תשעא] לפניך ממשק של המחלקה "תור-ביטול" (UndoQueue):

המחלקה מגדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה ולהוצאה של ערכים שלמים וגדולים מ-0.

שם הפעולה	תיאור
UndoQueue()	הפעולה בונה תור-ביטול ריק.
bool IsEmpty()	הפעולה מחזירה true אם התור-ביטול בנוכחי ריק, אחרת – הפעולה מחזירה false.
void Insert(int x)	הפעולה מכניסה את הערך x לסוף התור-ביטול הנוכחי.
int Remove()	הפעולה מוציאה את הערך שבראש התור-ביטול הנוכחי, ומחזירה אותו. <u>הנחה</u> : התור-ביטול הנוכחי אינו ריק.
int Head()	הפעולה מחזירה את הערך של האיבר שבראש בתור-ביטול בלי להוציאו. <u>הנחה</u> : התור-ביטול הנוכחי אינו ריק.
void Undo()	הפעולה מבטלת את פעולת ה-Remove או את פעולת ה-Insert האחרונה שבוצעה, ומשחזרת את התור-ביטול כפי שהיה לפני ביצוע הפעולה. אם אין פעולות Remove או Insert שבוצעו, הפעולה אינה עושה דבר. <u>שים לב</u> : הפעלה n פעמים של הפעולה Undo, תחזיר את התור-ביטול למצב שבו הוא היה לפני ביצוע n פעולות remove או Insert האחרונות שטרם בוטלו.

א. לפניך קטע קוד.

```
UndoQueue q = new UndoQueue();
q.Insert(1);
q.Insert(2);
q.Insert(3);
q.Remove();
q.Insert(4);
q.Undo();
q.Undo();
```

הראה את מצב התור-ביטול לאחר הביצוע של כל הוראה בקטע הקוד.

- ב. כתוב את כותרת המחלקה **UndoQueue**, ואת התכונות שלה. רשום תיעוד לכל תכונה.
- ג. ממש את הפעולות **Remove**, **Insert**, ו-**Undo** המוצגות בממשק המחלקה **UndoQueue**. אתה יכול להשתמש בפעולות הממשק האחרות של **UndoQueue** בלי לממש אותן.

## אוספים

**32.** [בגרות תשסח] בעלים של חנות למשחקי מחשב רוצים לבנות מאגר מידע על 100 המשחקים המוצעים למכירה בחנות. מאגר המידע יאפשר ללקוחות מידע על המשחקים האלה, והוא מתעדכן בכל פעם שלקוח משתמש במשחק. בעבור כל משחק נשמר המידע הזה:

- קוד המשחק.
  - שם המשחק.
  - הגיל המינימלי של שחקן שהמשחק מתאים לו.
  - מספר העותקים של המשחק שנמכרו בחנות.
  - מספר הפעמים הכולל שלקוחות שיחקו במשחק.
  - מספר הפעמים הכולל שבהן השימוש במשחק נפסק בגלל תקלה.
- א. כתוב ייצוג למחלקה **משחק** – **Game** ולמחלקה **מאגר מידע על משחקים** – **GamaData** שיאפשרו לשמור את כל המידע הנדרש. לכל מחלקה יש לרשום כותרת ותכונות.
- ב. לפניך שתי משימות i-ii. כתוב כותרות לפעולות שיאפשרו לבצע כל אחד מהמשימות. לכל פעולה יש לכתוב כותרת מלאה ותיעוד של הפרמטרים, ולציין את המחלקה שבה היא מוגדרת. הנח שהנתונים מיוצגים על פי הייצוג שכתבת בסעיף א.

### המשימות:

- i. עדכון מאגר המידע, על פי קוד משחק, כאשר השימוש במשחק נפסק בגלל תקלה.
  - ii. החזרת מספר המשחקים שנמכרו ביותר מ-k עותקים. k מספר שלם וגדול מ-0.
- ג. המשחקים מדורגים באמצעות כוכבים, לפי אחוז הפעמים שבהן השימוש במשחק נפסק בגלל תקלה (מבין כל הפעמים שלקוחות שיחקו במשחק):

\*\*\*\* פחות מ-5%

\*\*\* 5% או יותר, אך פחות מ-25%

\*\* 25% או יותר, אך פחות מ-50%

\* 50% או יותר

(1) כתוב פעולה פנימית במחלקה **Game**, שמחזירה את מספר הכוכבים המציין את דירוג המשחק.

(2) כתוב פעולה פנימית במחלקה **GamaData** שתקבל גיל מינימלי **minAge** ותחזיר רשימה של כל הקודים של המשחקים אשר מתאימים לשקחנים החל מגיל זה וגם מדורגים כמשחקים של 4 כוכבים.

הנח כי הוגדרו עבור כל תכונה פעולות **Get** ו-**Set**.

33. [בגרות תשס"ט] יומן אלקטרוני לניהול פגישות מכיל את הימים של שנה אחת. כל יום מיוצג על ידי :

- תאריך, הכולל חודש ויום בחודש.
  - רשימת הפגישות באותו ביום. בעבור כל פגישה נשמרים : שעת התחלת הפגישה, תוכן הפגישה, ומשך הפגישה בדקות. משך הפגישה לא חייב להיות שעות שלמות.
- בכל יום יכולות להיות פגישות בין 8:00 בבוקר ל-20:00 בערב. כל פגישה מתחילה בשעה עגולה. הפגישות מסתיימות לכל המאוחר ב-20:00 בערב. לפניך תיאור חלקי של המחלקה **פגישה – Meeting**.

<b>Meeting</b>	
private int startHour;	// שעת התחלת הפגישה
private int minutes;	// משך זמן הפגישה בדקות
private string content;	// תוכן הפגישה
public Meeting (int startHour, int minutes, string content)	
public int GetStartHour()	
public int GetMinute()	
public string GetContent()	

לפניך חלק מממשק המחלקה יום ביומן – **DayInSchedule** :

<b>DayInSchedule</b>	
public int GetDay()	
public int GetMounth()	
public Node<int> GetFreeHours()	הפעולה מחזירה הפנייה לתחילת רשימה של השעות הפנויות ביום בין 8:00 ל-19:00 (שעות שבהן לא מתקיימות פגישות). כל איבר ברשימה מכיל שעה עגולה אחת. הרשימה ממוינת בסדר עולה.
public bool CanStart(int startHour, int minutes)	הפעולה מקבלת שעת התחלה ומשך זמן הפגישה. הפעולה מחזירה true אם אפשר להתחיל את הפגישה בשעה המבוקשת, אחרת – הפעולה מחזירה false. אפשר להתחיל פגישה בשעה פנויה, בתנאי שיש אחריה די שעות פנויות כדי לסיים את הפגישה (בהתאם למשך הפגישה).

א. כתוב את כותרת המחלקה **DayInSchedule** ואת התכונות שלה.

ב. ממש את הפעולה CanStart המוצגת בממשק המחלקה **DayInSchedule**. אתה יכול להשתמש בפעולות האחרות של המחלקות Meeting ו-**DayInSchedule** בלי לממש אותן.

ג. ממש פעולה חיצונית שתקבל רשימה של ימים ביומן, ופגישה. הפעולה תדפיס את החודש והיום בחודש של כל אחד מהימים ברשימה, שבהם אפשר לשבץ את הפגישה. כותרת הפעולה :

public static void PrintAvailableDay(Node<DayInSchedule> lst, Meeting m)

34. [בגרות תשע] גולשים באתר "פטפט" יכולים לשלוח הודעות זה לזה. לכל גולש מוקצת תיבת הודעות

בגודל 100KB. בכל תיבה מאוחסנות ההודעות שהגולש קיבל. לכל הודעה מוקצה מספר סידורי. ככל שההודעה חדשה יותר המספר הסידורי שלה גבוה יותר. כל הודעה מכילה את שם השולח, תוכן ההודעה, מספר סידורי גודל ההודעה בקילו-בית. גודל ההודעה הוא מספר שלם. לתיבת ההודעות יש סל אשפה שהוא חלק מהתיבה. הודעות שנמצאות בתיבת ההודעות ואינן בסל האשפה נקראות הודעות פעילות.



גולש יכול להעביר הודעות פעילות לסל האשפה או למחוק הודעות שנמצאות בסל האשפה. העברת הודעות לסל האשפה אינה גורמת לשינוי המספר הסידורי שלהן. הודעות שנמחקו מסל האשפה אינן קיימות יותר בתיבה. מחיקת הודעות אינה גורמת לשינוי המספר הסידורי של ההודעות שנשארו בתיבה.

לפניך חלק מממשק המחלקה **תיבת הודעות – MessageBox**.

<b>MessageBox</b>	
public MessageBox(string owner)	פעולה בונה המקבלת את שם בעל התיבה ויוצרת תיבה ריקה בגודל 100 קילו-בית.
public bool AddMessage(Message m)	פעולה המוסיפה הודעה חדשה m מטיפוס Message. אם אין די מקום בתיבה להודעה חדשה, יימחקו מסל האשפה ההודעות בעלות המספרים הסידוריים הנמוכים ביותר עד שיהיה די מקום להודעה חדשה. אם אין די מקום בתיבה ואין די הודעות למחוק בסל האשפה, ההודעה <u>לא</u> תתווסף לתיבה, <u>ולא</u> יימחקו הודעות מסל האשפה. הפעולה תחזיר true אם ההודעה הוספה לתיבה, אחרת הפעולה תחזיר false. <u>הנחות:</u> ל-m יש מספר סידורי מעודכן. גודל ההודעה קטן מגודל תיבת ההודעות.
public int RemoveFromBin()	פעולה המוחקת את ההודעה בעל המספר הסידורי הקטן ביותר בסל האשפה ומחזירה את גודלה בקילו-בית.
public void EmptyBin()	פעולה המוחקת את כל ההודעות מסל האשפה.
public int GetActiveSize()	פעולה המחזירה את סך כל הגודל-בית שתוספות כל ההודעות הפעילות.
public int GetBinSize()	פעולה המחזירה את סך כל הגודל בקילו-בית שתופסות כל ההודעות בסל האשפה.

א. כתוב את כותרת המחלקה **Message** המייצגת הודעה ב-**MessageBox**, ואת התכונות שלה. רשום תיעוד לכל תכונה.

ב. כתוב את כותרת המחלקה **MessageBox** ואת התכונות שלה. רשום תיעוד לכל תכונה.

ג. ממש את הפעולה **AddMessage** המוצגת בממשק המחלקה **MessageBox**. אתה יכול להשתמש בפעולות האחרות של המחלקה **MessageBox** מבלי לממש אותן. אתה יכול להניח שלכל התכונות שרשמת בסעיף א השייכות למחלקה **Message** יש פעולות **Get** ו-**Set**.

ד. לאחר שגולשים "פטפונים" רבים ביקשו להגדיל את התיבה, הוחלט שפעם ביום תעבור המערכת על כל התיבות של הוגלשים ותקטין או תגדיל את התיבות לפי הפעילות שלהם. תיבה שהתפוסה שלה יותר מ-80% תוגדל פי 2, ותיבה שהתפוסה שלה פחות מ-30% תוקטן בחצי.

1) האם לאור החלטה זו תצטרך לשנות את תכונות המחלקה **MessageBox** שכתבת בסעיף ב?

אם כן – הסבר מדוע, וכתוב את התכונות החדשות, ואם לא – הסבר מדוע.

2) האם לאור החלטה זו תצטרך לשנות את הפעולה שכתבת בסעיף ג? אם כן – הסבר אילו שינויים יש לעשות, ואם לא – הסבר מדוע.

35. [בגרות תשעב] **קבוצה ממשית (RealSet)** היא אוסף של מספרים ממשיים בלי חזרות ובלי סדר מחייב של הערכים. לפניך ממשק חלקי של המחלקה **RealSet**.

שם הפעולה	תיאור
RealSet()	הפעולה בונה <b>קבוצה ממשית</b> ריקה.
int Size()	הפעולה מחזירה את מספר האיברים בקבוצה.
void Insert(double num)	הפעולה מוסיפה לקבוצה אם המספר num, אם הוא לא נמצא בקבוצה.
void Remove(double num)	הפעולה מוחקת את המספר num, אם הוא נמצא בקבוצה.
double FindBiggest()	הפעולה מחזירה את המספר הגדול בקבוצה. הנח שהקבוצה אינה ריקה.

א. כתוב פעולה חיצונית בשם Clone, שתקבל **קבוצה ממשית** rs לא ריקה, ותחזיר **קבוצה ממשית** חדשה שתכיל את כל האיברים שיש ב-rs. לאחר סיום ביצוע הפעולה, על rs להכיל את כל המספרים שהיא הכילה לפני ביצוע הפעולה ורק אותם.

ב. כתוב פעולה חיונית בשם BuildNeg, שתקבל קבוצה ממשית rs לא ריקה, ותחזיר **קבוצה ממשית** שתכיל רק את המספרים השליליים שנמצאים ב-rs. לאחר סיום ביצוע הפעולה, על rs להכיל את כל המספרים שהיא הכילה לפני ביצוע הפעולה, ורק אותם. השתמש בפעולה שכתבת בסעיף א. אתה יכול להשתמש בפעולות המחלקה **RealSet** בלי לממש אותן.

36. [בגרות תשעב] המשחק "שחק נא" הוא משחק לשחקן אחד בשני שלבים. השחקן מקבל 52 קלפים מעורבבים, שונים זה מזה. על כל קלף יש מספר בין 1 ל-13, וציר של אחת מארבע צורות. הצורות מיוצגות על ידי מספרים 1 עד 4.

בשלב הראשון השחקן מחלק את הקלפים לארבע ערמות על פי הצורה של הקלף, כך שבכל ערמה יש קלפים עם אותה צורה. הקלפים מונחים זה על גבי זה.

בשלב השני השחקן מגריל מספר בין 1 ל-4 המייצג צורה של קלף. השחקן פונה לערמה שעל כל הקלפים בה נמצאת הצורה שמיוצגת על ידי המספר שהוגרל. הוא מרים את הקלף שבראש הערמה, ומעביר אותו לערמה חמישית. שלב זה יתבצע עד שהמשחק יסתיים.

**המשחק מסתיים** כאשר מוגרל מספר המייצג ערמה שאין בה קלפים.

**ניצחון** הוא מצב שבו סכום המספרים שעל הקלפים בערמה החמישית מתחלק ב-100 ללא שארית. לאחר שהמשחק מסתיים בודקים אם יש ניצחון.

לצורך מימוש המשחק הוגדרו המחלקות: **Card** המייצגת קלף אחד, **Deck** המייצגת את חמש הערמות הנדרשות במשחק, ו-**Test** המנהלת את המשחק.

כמו כן נכתבה במחלקה Test הפעולה `public static bool Game(Card[] cards)`, המקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיומו. הפעולה מחזירה true אם המשחק הסתיים בניצחון, ואחרת – מחזירה false.

לפניך המחלקה **Card** המייצגת קלף שעליו מספר (value) בין 1-13 וצורה (shape) בין 1 ל-4.

```

public class Card
{
    private int value;
    private int shape;
    public Card(int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int GetValue() { return this.value; }
    public int GetShape() { return this.shape; }
}

```

- א. כתוב את כותרת המחלקה **Deck** ואת התכונות שלה.
  - ב. כתוב במחלקה **Deck** את כותרות הפעולות האלה:
    - פעולה בונה המגדירה את חמש הערמות להיות ריקות.
    - פעולה Insert המקבלת קלף ומוסיפה אותו לערמה הנכונה על פי השלב הראשון במשחק.
    - פעולה Move המגרילה מספר המייצג צורה. אם ערמת הקלפים שצורתה הוגרלה היא ריקה, הפעולה תחזיר false, אחרת - הפעולה תעביר לערמה החמישית את הקלף שבראש הערמה שמספרה הוגרל ותחזיר true.
  - שים לב: בתת-סעיף זה אין צורך לממש את הפעולות.
  - ג. ממש את הפעולה הבונה במחלקה **Deck**.
  - ד. ממה את הפעולה Sum במחלקה **Deck**.
  - ה. ממה את הפעולה Game במחלקה **Test**.
- אתה יכול להשתמש בפעולות Insert ו-Move של המחלקה **Deck** בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.
37. [בגרות תשעד] המחלקה **Collec** היא אוסף של מספרים שלמים וגדולים מ-0. לאוסף זה אפשר להוסיף איבר רק אם אין באוסף איבר אחר גדול ממנו.
- א. כתוב את כותרת המחלקה **Collec** ואת התכונות שלה.
  - ב. כתוב פעולה בונה במחלקה **Collec** שתקבל מספר שלם n גדול מ-0, ותיצור אוסף שיש איבר אחד שערכו n.
  - ג. ממש במחלקה **Collec** פעולה Add, שתקבל מספר שלם וגדול מ-0 ותוסיף אותו לאוסף, אם אפשר. אם המספר צורף לאוסף, הפעולה תחזיר true, אחרת – יוחזר false.
  - ד. ממש במחלקה **Collec** פעולה Small, שתחזיר את המספר הקטן ביותר באוסף.
  - ה. ממש במחלקה **Collec** פעולה Smallest בשם c, שתקבל עצם c מטיפוס **Collec**, ותחזיר את המספר הקטן ביותר מבין שני האוספים – האוסף הנוכחי ו-c.

38.

[בגרות תשעה] בחברה "עבודה יעילה" יש משימות : בהולות, דחופות ורגילות. משימות בהולות מתבצעות ראשונות אחריהן מתבצעות משימות דחופות ולבסוף משימות רגילות. בחברה יש מערכת ממוחשבת **סדרן משימות**. ההכנה של המשימות ל**סדרן משימות** נעשית על פי הכללים האלה :

- משימה בהולה חדשה תוכנס לפני כל המשימות – הבהולות, הדחופות והרגילות – שנמצאות כרגע ב**סדרן משימות**.

- משימה דחופה חדשה תוכנס אחרי כל המשימות הבהולות ו**לפני** המשימות הדחופות והרגילות שנמצאות כרגע ב**סדרן משימות**.

- משימה רגילה חדשה תוכנס אחרי כל המשימות – הבהולות, הדחופות והרגילות – שנמצאות כרגע ב**סדרן משימות**.

הוצאה לביצוע של משימה מ**סדרן משימות** נעשית על פי הסדר שנוצר ב**סדרן המשימות**.

נתונה המחלקה משימה – Task, שיש לה שתי תכונות : Content – מחרוזת שהיא תיאור המשימה, ו- code – מספר שלם המייצג את המשימה : 1 מייצג משימה בהולה ; 2 מייצג משימה דחופה ; 3 מייצג משימה רגילה. הנח שלכל תכונה יש פעולות Get ו-Set.

יש לממש את המחלקה **סדרן משימות** בעזרת מספר כלשהו של מחסניות ותורים, כך שסיבוכיות זמן הריצה של פעולות ההכנסה ל**סדרן משימות** ופעולות ההוצאה מ**סדרן משימות** תהיה  $O(1)$ .

א. כתוב את כותרת המחלקה **סדרן משימות** ואת התכונות שלה כתוב תיעוד לכל תכונה.

ב. כתוב פעולה בונה ללא פרמטרים של המחלקה **סדרן משימות**.

ג. כתוב במחלקה **סדרן משימות** פעולה שתקבל משימה ותכניס אותה ל**סדרן משימות**, בהתאם לכללים המתוארים בתחילת השאלה.

ד. כתוב במחלקה **סדרן משימות**, פעולה שתוציא את המשימה הבאה לביצוע ותחזיר אותה. אם אין משימה ב**סדרן משימות** הפעולה תחזיר null.

39. [בגרות תשעו] לפניך המחלקה טבעת – Ring שלה שתי תכונות: גודל הטבעת מטיפוס מחרוזות ("S") – טבעת קטנה, "L" – טבעת גדולה; מספר שלם המייצג את צבע הטבעת.

```
public class Ring
{
    private string size;
    private int color;
    public Ring()
    {
        this.size = "L";
        this.color = 0;
    }
    public Ring(string str, int c)
    {
        this.size = str;
        this.color = c;
    }
    public string GetSize() { return this.size; }
    public int GetColor() { return this.color; }
}
```

לפניך ממשק המחלקה מוט – Pole

הפעולה	תיאור
public Pole()	פעולה הבונה מוט ריק. סיבוכיות זמן הריצה של הפעולה היא $O(1)$
public void Add(Ring r)	פעולה המכניסה טבעת r לראש המוט. סיבוכיות זמן הריצה של הפעולה היא $O(1)$
public Ring Remove()	פעולה המחזירה את הטבעת שבראש המוט. בנוסף, הפעולה מוציאה את הטבעת מראש המוט. סיבוכיות זמן הריצה של הפעולה היא $O(1)$
public bool IsEmpty()	אם המוט ריק הפעולה מחזירה true, אחרת – היא מחזירה false. סיבוכיות זמן הריצה של הפעולה היא $O(1)$
public void Sort()	פעולה המסדרת את הטבעות שעל המוט כך: הטבעות הגדולות "מונחות" בתחתית המוט והטבעות הקטנות מעליהן.

- א. ממש את הפעולה Sort() שבמחלקה Pole. אתה יכול להשתמש בפעולות הנוספות של המחלקה Pole בלי לממש אותן. בתשובתך השתמש רק בפעולות המחלקות Pole ו-Ring.
- ב. מבי סיבוכיות זמן הריצה של הפעולה שמימשת בסעיף א? נמק את תשובתך.
40. [בגרות תשעז] לפניך הגדרות של חמש פעולות הפועלות על מבנה נתונים כלשהו.

- insert(x)	פעולה המכניסה למבנה איבר שערכו x מטיפוס שלם.
- showMin()	פעולה המחזירה את הערך הנמוך ביותר במבנה, בלי לשנות את המבנה.
- getMax()	פעולה המחזירה את האיבר שערכו הוא הגדול ביותר במבנה, ומוציאה אותו מן המבנה. אם יש יותר מאיבר אחד כזה, הפעולה תחזיר ותוציא את זה שמופיע ראשון.
- exist(x)	פעולה בוליאנית המחזירה true אם האיבר שערכו x קיים במבנה, אחרת – הפעולה מחזירה false.
- div7()	פעולה בוליאנית המחזירה true אם קיין במבנה איבר שערכו מתחלק ב-7 בלי שארית. אחרת – הפעולה מחזירה false.

נרצה להציע מבני נתונים העומדים בדרישות סיבוכיות שונות למימוש פעולות מתוך 5 הפעולות שהוגדרו. דוגמה: רוצים להציע מבנה נתונים שאפשר לבצע עליו את הפעולות insert, showMin, בסיבוכיות  $O(1)$ , ואת הפעולות getMax, exist, בסיבוכיות  $O(n)$ . לשם כך נגדיר את מבנה הנתונים ונסביר כיצד ימומשו הפעולות. שים לב: במבנה זה אין צורך להתייחס לפעולה div7.

מבנה נתונים מתאים מורכב מ- :

- רשימה מקושרת דו-כיוונית lst מטיפוס שלם.
- מצביע על האיבר המינימלי שהוכנס לרשימה, min.

הפעולות יבוצעו כך :

הפעולה	הסבר כיצד תמומש	נימוק מדוע המימוש עומד בדרישות הסיבוכיות
<b>insert(x)</b>	הכנסת האיבר x לראש הרשימה. אם האיבר קטן מן המינימום עד כה, עדכון המצביע min כך שיצביע על האיבר החדש.	הכנסת איבר לרשימה – $O(1)$ אם הוכנס איבר שקטן מן המינימום : עדכון מצביע לאיבר המינימלי – $O(1)$ . אם הוכנס איבר ראשון – הכנסה לראש הרשימה ב- $O(1)$ ועדכון המצביע לאיבר הראשון ברשימה שהוא גם המינימלי. סך הכול – $O(1)$
<b>showMin()</b>	החזרת הערך של האיבר שעליו מבציע min.	החזרת הערך – $O(1)$
<b>exist(x)</b>	מעבר על הרשימה lst וחיפוש האיבר שערכו x.	במקרה הגרוע מעבר על כל הרשימה – $O(n)$
<b>getMax(x)</b>	מעבר על הרשימה lst חיפוש האיבר שערכו מקסימלי והוצאתו מן הרשימה.	חיפוש האביר שערכו מקסימלי $O(n)$ הוצאתו מן הרשימה $O(1)$

לפניך שני סעיפים א-ב. בעבור כל אחד מן הסעיפים עליך להציע מבנה נתונים מתאים העומד בדרישות המפורטות בסעיף. המבנה יכול להיות מורכב משילוב של כמה מבנים וטיפוסים שלמדת. לכל אחת מהפעולות הסבר כיצד תממש אותה, ונמק מדוע המימוש עומד בדרישות (כפי שהוצג בטבלת שבדוגמה). אין צורך לממש את הפעולות.

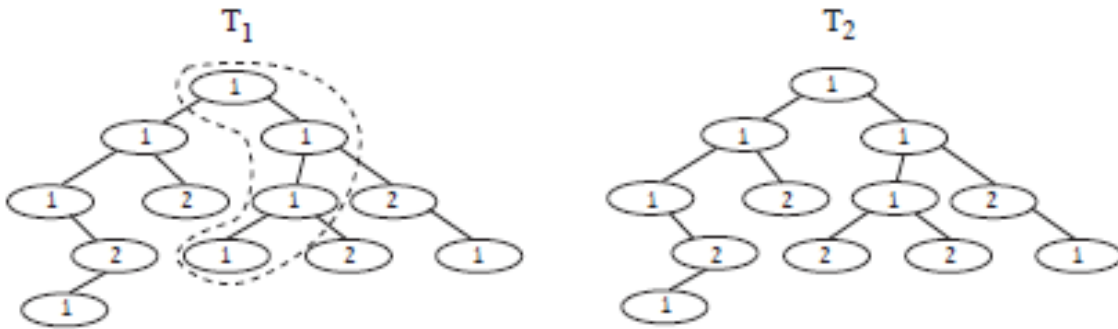
א. ביצוע הפעולות exist , insert בסיבוכיות  $O(n)$ , וביצוע הפעולות showMin , getMax בסיבוכיות  $O(1)$ .

ב. ביצוע הפעולות gatMax , insert בסיבוכיות  $O(n)$ , וביצוע הפעולה div7 בסיבוכיות  $O(1)$ .

## עצים

41. [בגרות תשסז] עץ "דו-מספרי" הוא עץ בינארי לא ריק שהערכים בצמתים שלו הם מספרים 1 או 2. על עץ "דו-מספרי" מוגדרת פעולה **מסלול-אחיד** המחזירה true אם קיים בעץ מסלול המתחיל בשורש העץ ומסתיים באחד העלים שלו, וכל ערכי הצמתים בו זהים. אם לא קיים מסלול כזה, הפעולה תחזיר false. דוגמאות:

בעבור העץ T1 הפעולה **מסלול-אחיד** תחזיר true. בעבור העץ T2 הפעולה **מסלול-אחיד** תחזיר false.

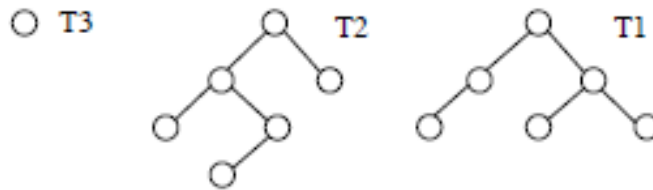


ממש את הפעולה מסלול-אחיד בעבור עץ דו-מספרי t.

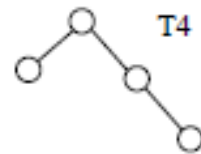
כותרת הפעולה: `public static bool OnePath(BinNode<int> t)`

42. [בגרות תשסט] עץ בינארי ייקרא עץ **ימין-שמאל** אם לכל צומת בעץ אשר יש לו בן ימני, יש לו גם בן שמאלי. לדוגמה:

העצים T1, T2, T3 הם עצי ימין-שמאל



והעץ T4 אינו עץ ימין-שמאל



כתוב פעולה חיצונית שתקבל עץ בינארי T שהצמתים שלו הם מטיפוס שלם. הפעולה תחזיר true אם העץ הוא **ימין-שמאל**, אחרת – הפעולה תחזיר false.

43. [בגרות תשסט] נתון עץ בינארי שהצמתים שלו מכילים ערכים מטיפוס תו. אם סורקים את העץ T בסדר תחילי (preorder), אזי סדר הערכים המתקבל מביקור במצתים (משמאל לימין) הוא:

X, A, I, O, N, Y, T, D, S

אם סורקים את העץ T בסדר תוכי (inorder), אזי הערכים המתקבלים מביקור בצמתים (משמאל לימין) הוא:

I, N, O, A, X, D, T, S, Y

צייר את העץ T, ורשום את סדר הערכים המתקבל מביקור במצתים אם סורקים את העץ בסדר סופי (postorder).

## .44 [בגרות תשעא]

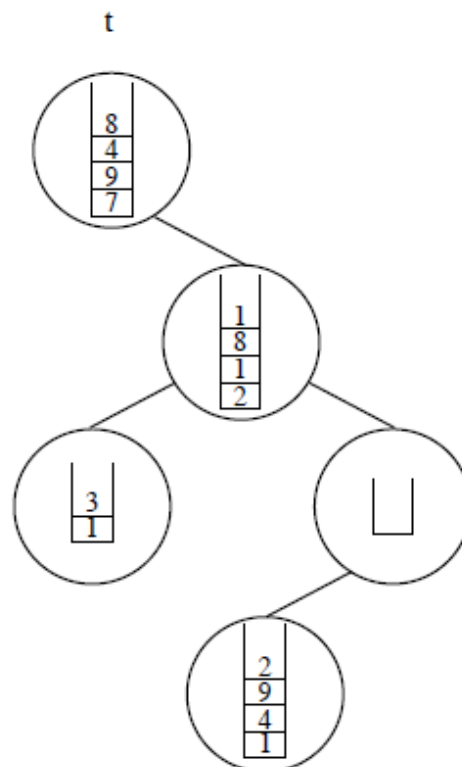
- א. לפניך כותרת של פעולה: `public static void Leaves (BinNode<int> t, Stack<int> s)`. הפעולה מקבלת עץ בינארי לא ריק  $t$  של מספרים שלמים, ומחסנית ריקה  $s$  של מספרים שלמים. הפעולה מכניסה למחסנית את ערכי כל העלים של העץ  $t$ , על פי סדר סריקה מימין למשאל. ממש את הפעולה.
- ב. כתוב פעולה בוליאנית שתקבל 2 עצים בינאריים לא ריקים של מספרים שלמים, ותחזיר `true` אם מתקיימים שני התנאים האלה:
- יש להם אותו מספר עלים.
  - על פי סדר סריקה מימין לשמאל ערכי העלים שווים.
- אחרת – הפעולה תחזיר `false`. עליך להשתמש בפעולה שממשת בסעיף א.

## .45 [בגרות תשעב]

- א. כתוב פעולה חיצונית שתקבל עץ בינארי  $t$  לא ריק, שבו כל צומת מכיל מחסנית של מספרים שלמים גדולים מ-0. הפעולה תחזיר מחסנית. בעבור כל צומת בעץ  $t$  יוכנס איבר למחסנית המוחזרת באופן הזה: אפ במחסנית שבצומת יש איבר אחד, יוכנס ערכו למחסנית שתוחזר. אם במחסנית שבצומת יש שני איברים, יוכנס סכומם למחסנית שתוחזר. אם המחסנית שבצומת ריקה, יוכנס 0 למחסנית שתוחזר. בכל מקרה אחר יוכנס למחסנית שתוחזר הסכום של שלושת האיברים העליונים של המחסנית שבצומת. סדר האיברים במחסנית שתוחזר יהיה לפי סריקה בסדר תוכי (`inorder`) של צומתי העץ  $t$ . אין צורך לשמור את תוכן המחסניות שבצומתי העץ. לדוגמה בעבור העץ  $t$ :

המחסנית שתוחזר:

0
15
10
4
21



- ב. מהי סיבוכיות זמן הריצה של הפעולה שכתבת סעיף א? נמק את תשובתך.



46. [בגרות תשעג] עץ **טרינארי** הוא עץ שיש בו לכל היותר שלושה בנים לצומת, בן שמאלי, בן אמצעי ובן ימני. לפניך ממשק חלקי שהמחלקה צומת טרינארי `TriNode<T>`. המחלקה מגדירה **צומת טרינארי** שבו ערך מטיפוס `T` ושלוש הפניות לצמתים טרינאריים.

שם הפעולה	תיאור
<code>TriNode(T x)</code>	הפעולה בונה <b>צומת טרינארי</b> שערכו יהיה <code>x</code> , וערך שלוש ההפניות יהיה <code>null</code> .
<code>TriNode(TriNode&lt;T&gt; left, TriNode&lt;T&gt; middle, TriNode&lt;T&gt; right, T x)</code>	הפעולה בונה צומת טרינארי שערכו יהיה <code>x</code> , <code>left</code> – הפניה לבן שמאלי, <code>middle</code> – הפניה לבן אמצעי ו- <code>right</code> – הפניה לבן ימני. ערכי הפניות יכולות להיות <code>null</code> .
<code>T GetValue()</code>	הפעולה מחזירה את הערך של הצומת הטרינארי.
<code>void SetValue(T x)</code>	הפעולה משנה את הערך השמור בצומת הטרינארי ל- <code>x</code> .
<code>TriNode&lt;T&gt; GetLeft()</code>	הפעולה מחזירה את הבן השמאלי של הצומת הטרינארי. אם אין בן שמאלי, הפעולה מחזירה <code>null</code> .
<code>TriNode&lt;T&gt; GetMiddle()</code>	הפעולה מחזירה את הבן האמצעי של הצומת הטרינארי. אם אין בן אמצעי, הפעולה מחזירה <code>null</code> .
<code>TriNode&lt;T&gt; GetRight()</code>	הפעולה מחזירה את הבן הימני של הצומת הטרינארי. אם אין בן ימני, הפעולה מחזירה <code>null</code> .

א. כתוב פעולה חיצונית `Big` שתקבל עץ טרינארי `t` של מספרים שלמים וגדולים מ-0, ותחזיר את המספר הגדול ביותר בעץ. אם העץ ריק, הפעולה תחזיר -1.

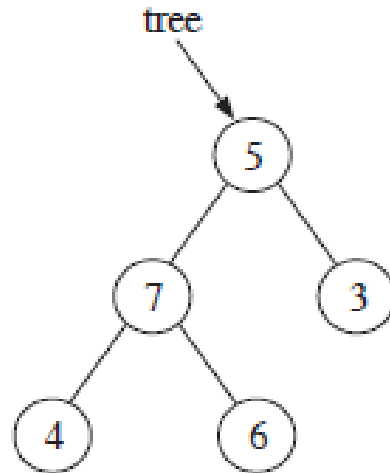
ב. לפניך כותרת של פעולה: `public static bool NoThree(TriNode<T> tr)`. הפעולה מקבלת עץ טרינארי `tr` של מספרים שלמים ומחזירה `true`, אם בכל אחד מהצמתים שבעץ יש לכל היותר שני בנים. אחרת – הפעולה מחזירה `false`. אם העץ ריק הפעולה מחזירה `true`. ממש את הפעולה.

הערה: אתה יכול להשתמש בפעולות המחלקה **TriNode** בלי לממש אותן.

47. [בגרות תשעג] לפניך הפעולה Amir.

```
public static void Amir(BinNode<int> t, int x)
{
    if (t != null)
    {
        Amir(t.GetLeft(), x + 1);
        if ((t.GetLeft() == null) && (t.GetRight() == null))
            t.SetValue(x);
        Amir(t.GetRight(), x + 1);
    }
}
```

לפניך העץ tree.



א. רשום את העץ tree שיתקבל לאחר הפעלת הזימון: Amir(tree,4)

ב. מה מבצעת הפעולה Amir בעבור עץ כלשהו ו-0=x.

48. [בגרות תשעג] נתון עץ חיפוש בינארי t, המכיל לפחות שלושה צמתים שערכיהם: x, y, z. לפניך שתי טענות:

(1) אם x, y ו-z הם ערכי שלושה צמתים עוקבים בסריקה תוכנית (inorder) של t אז x=z.

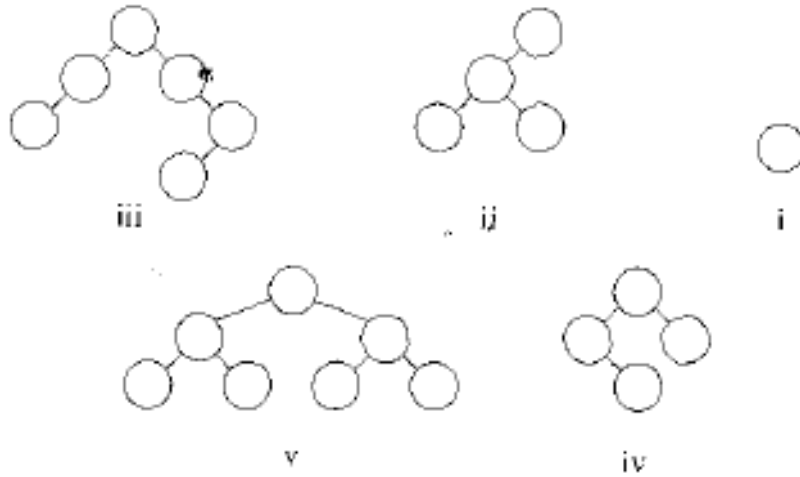
(2) אם x ו-y הם ערכים של שני עלים ב-t (משמאל לימין), אז x > y.

לכל אחת מהטענות (1) - (2) בחר באפשרות המתאימה מבין שלוש האפשרויות שלפניך:

- הטענה תמיד נכונה. נמק את תשובתך.
- הטענה תמיד אינה נכונה. נמק את תשובתך.
- הטענה נכונה בחלק מהמקרים. הבא דוגמה למקרה שבו הטענה נכונה, ודוגמה למקרה שבו הטענה אינה נכונה.

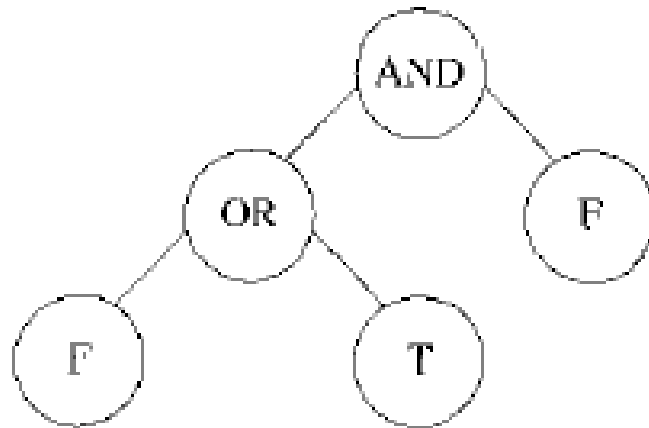
49. [בגרות תשעד] **עץ בינארי מאוזן** הוא עץ בינארי שבכל צומת שלו הערך המוחלט של הפרש הגבהים בין התת-עץ הימני לתת-עץ השמאלי הוא לכל היותר 1.

א. לפניך 5 עצים בינאריים i-v. לכל אחד מהעצים קבע אם הוא עץ בינארי מאוזן או אינו עץ בינארי מאוזן. אם העץ אינו עץ בינארי מאוזן, העתק אותו למחברתך וסמן X בצומת שמפר את האיזון.

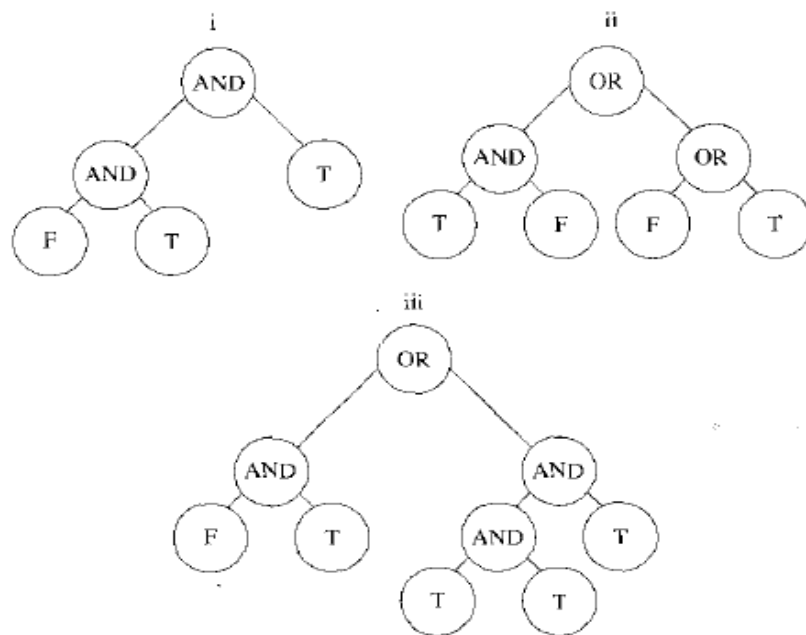


ב. כתוב פעולה שתקבל עץ בינארי כלשהו, ותחזיר true או הוא עץ בינארי מאוזן, אחרת – הפעולה תחזיר false. אם העץ ריק – הפעולה תחזיר נ"ל.

50. [בגרות תשעה] **עץ ביטוי בוליאני** הוא עץ בינארי לא ריק מטיפוס מחרוזות, המייצג ביטוי בוליאני. בכל עלה שלו נמצא אחת מן המחרוזות: "T" או "F". המחרוזת "T" מייצגת true, והמחרוזת "F" מייצגת false. בכל צומת שאינו עלה נמצאת אחת מן המחרוזות: "AND" או "OR". המחרוזת "AND" מייצגת את הפעולה הבוליאנית "וגם", והמחרוזת "OR" מייצגת את הפעולה הבוליאנית "או". לכל צומת שאינו עלה יש שני בנים. כדי לחשב את הביטוי הוליאני שהעץ מייצג, מפעילים את הפעולה הבוליאנית שבצומת שאינו עלה על הערכים המתקבלים מהתת-עצים השמאלי והימני של צומת זה. לדוגמה: הביטוי הבוליאני שמייצג את העץ שלפניך הוא  $((F \text{ OR } T) \text{ AND } (F))$  וערכו הוא false.

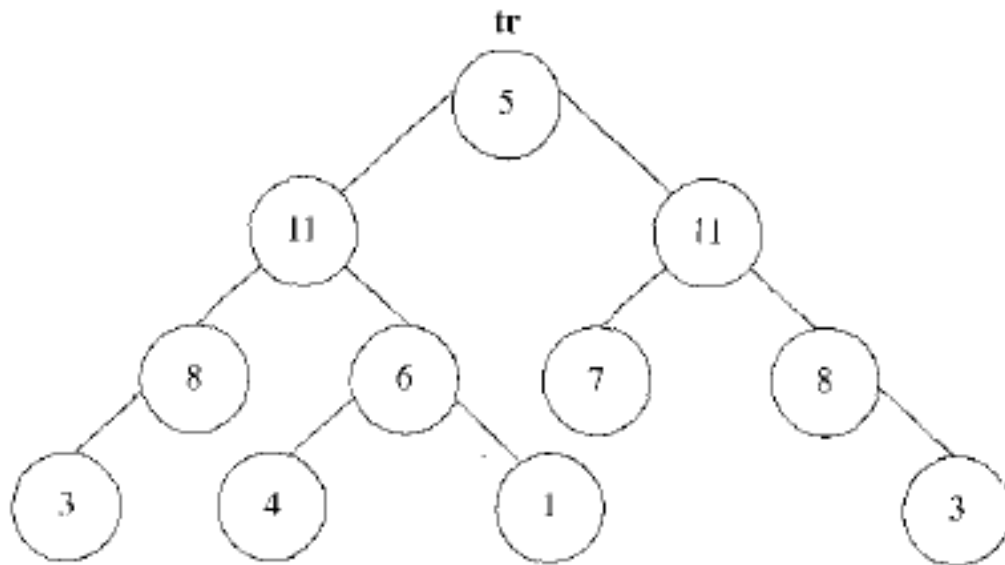


א. לפניך שלושה עצים i-iii שכל אחד מהם הוא עץ ביטוי בוליאני. לכל אחד מן העציים כותב את הביטוי הבוליאני שהוא מייצג, ואת הערך המתקבל ממנו.



ב. כתוב פעולה חיצונית שתקבל עץ ביטוי בוליאני, ותחזיר את הערך הבוליאני (true או false) של הביטוי שהעץ מייצג.

## 51. [בגרות תשעה] נתון עץ בינארי tr



לפניך הפעולה Wrap המשתמשת בפעולה Branch.

א. עקוב אחר ביצוע הפעולה Wrap בעבור העץ tr הנתון, ורשום את הערך שיוחזר.

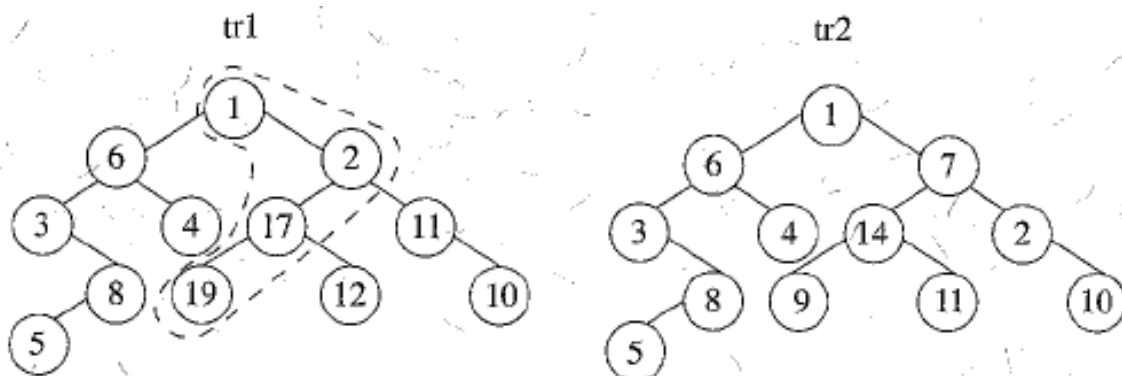
ב. הבא דוגמה לעץ בינארי שבעבורו Wrap תחזיר true ודוגמה לעץ בינארי שבעבורו הפעולה תחזיר false. על כל אחד מהעצים להכיל 5 צמתים בדיוק.

```
public static bool Branch(BinNode<int> t1, BinNode<int> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.GetValue() == t2.GetValue()) && Branch(t1.GetLeft(), t2.GetRight()));
}
```

## 52. [בגרות תשעו] עץ מספרים הוא עץ בינארי לא ריק מטיפוס שלם, שהערכים בצמתים שלו הם מספרים

שלמים וגדולים מ-0 השונים זה מזה. על עץ מספרים מוגדרת פעולה "מסלול-עולה", המחזירה true אם יש בעץ מסלול המתחיל בשורש העץ ומסתיים באחד העלים שלו, וערכי הצמתים ממוינים בסדר עולה מהשורש לעלה. אם אין מסלול כזה – הפעולה תחזיר false.

לדוגמה: בעבור עץ מספרים tr1 הפעולה "מסלול-עולה" מחזירה true. המסלול מוקף בקו שבור. בעבור עץ מספרים tr2 הפעולה "מסלול-עולה" מחזירה false.



ממש את הפעולה מסלול-עולה בעבור עץ מספרים tr.

כותרת הפעולה: `public static bool UpPath(BinNode<int> tr)`

## 53. [בגרות תשעז]

א. ממש פעולה חיצונית Exist. הפעולה תקבל עץ בינארי t מטיפוס שלם ומספר שלם x. הפעולה תחזיר true אם יש בעץ צומת שערכו x, אחרת – הפעולה תחזיר false. אם העץ ריק הפעולה תחזיר false.

ב. לפניך הפעולה Check(t1, t2). הפעולה מקבלת שני עצים בינאריים לא ריקים מטיפוס שלם, t1 ו-t2, ומחזירה את כל המספרים הנמצאים בעץ t1 ואינם נמצאים בעץ t2. הפעולה מזמנת פעולה נוספת המקבלת שלושה פרמטרים.

```
public static Node<int> Check(BinNode<int> t1, BinNode<int> t2)
{
    Node<int> first = new Node<int>(-1);
    first = Check(t1, t2, first);
    return first.GetNext();
}
```

ממש את הפעולה :

```
public static Node<int> Check(BinNode<int> t1, BinNode<int> t2, Node<int> list)
```

אתה יכול להשתמש בפעלה שמימשת בסעיף א.

ג. מהי סיבוכיות זמן הריצה של הפעולה שמימשת בסעיף ב? נמק את תשובתך.

## תכנות מונחה עצמים

### מעקב אחרי Main

1. [בגרות תשסו] לפניך עכק דקכ'קכ
2. קכ'דגכ
3. דגכ
4. דגכ

### המרות וזיהוי שגיאות

5. דכ
6. דקכ דגכ
7. דגכ

### ממשקים

8. דגכדגכ

### מימוש תכנות מערכות

9. דגלכמלדגכל"