

מבני נתונים ותכנות מונחה עצמים

הנדסאים וטכנאים – הנדסת תוכנה

הנחיות לבחינה

מועד הבחינה :
אביב תשפ"ה – 2025 – מועד א'
מספר השאלון : 97105
נספח ממשקים לבחינה JAVA
נספח ממשקים לבחינה C#

א. משך הבחינה : ארבע שעות וחצי.

ב. מבנה השאלון בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:

ומפתח ההערכה: מבחן ב-C# (עמוד 2)

מבחן ב-JAVA (עמוד 12)

בכל מבחן עשר שאלות.

חלק א' – 48 נקודות

שאלות 1–5 : יש לענות על שלוש שאלות בלבד. ערך כל שאלה 16 נקודות.

חלק ב' – 36 נקודות

שאלות 6–10 : יש לענות על שתי שאלות בלבד. ערך כל שאלה 18 נקודות.

חלק ג' – 16 נקודות

שאלות 11–12 : יש לענות על שאלה אחת בלבד. ערך השאלה 16 נקודות.

בסך הכול: 100 נקודות.

ג. חומר עזר 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית).

2. מותר לשימוש: קלסר אחד בלבד עם חומר ההרצאות. אין להוציא דפים מהקלסר.

אין לצרף ספרים או חוברות עם פתרונות.

ד. הוראות כלליות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות המבחן, ולוודא שהן מובנות.

2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת המבחן).

3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה.

לא ייבדקו תשובות עודפות.

4. יש לכתוב את התשובות במחברת הבחינה בעט בלבד, בכתב יד ברור.

5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף.

אין צורך להעתיק את השאלה עצמה.

6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק.

7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד.

8. יש להסביר בפירוט כל תוכנית שנכתבה, תוכנית ללא הסבר מפורט לא תזכה בניקוד.

9. אם לדעתכם חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לכם להמשיך בפתרון השאלה, נמקו את בחירתכם.

חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!

בהצלחה!

מבחן ב-C#**חלק א'**

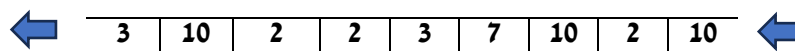
ענו על שלוש מבין השאלות 1–5 (ערך כל שאלה – 16 נקודות).

שאלה 1

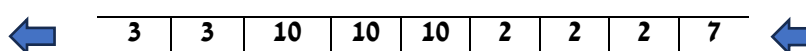
(14 נק') א. כתבו פעולה המקבלת תור של מספרים שלמים ומחזירה תור חדש המכיל את כל האיברים שבתור המקורי כך שהאיברים הזהים יהיו ברצף. סדר הרצפים בתור החדש אינו חשוב.

לדוגמה ,

עבור התור q הזה :



הפעולה תחזיר את התור (אחת התשובות האפשריות) :



כותרת הפעולה :

```
public static Queue<int> OrderQ(Queue<int> st)
```

(2 נק') ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

שאלה 2

(14 נק') א. נתונה מחסנית של מספרים שלמים. כל איבר במחסנית מופיע פעמיים בדיוק. המרחק בין שני איברים במחסנית הוא כמות האיברים הנמצאים ביניהם.

כתבו פעולה המחזירה את המרחק הגדול ביותר בין שני איברים זהים במחסנית.

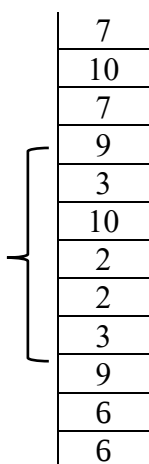
לדוגמה :

עבור המחסנית הבאה הפעולה תחזיר 5 (המרחק בין שני איברים שערכם 9).

כותרת הפעולה :

```
public static int MaxDistance(Stack<int> st)
```

(2 נק') ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

**שאלה 3**

(14 נק') א. שרשרת נקראת "שרשרת מושלמת" אם כל ערך חיובי שבה מופיע מספר זוגי של פעמים וכל ערך שלילי שבה מופיע מספר אי-זוגי של פעמים.

כתבו פעולה המקבלת הפניה לחוליה הראשונה בשרשרת של מספרים שלמים. הפעולה תבדוק אם השרשרת היא "שרשרת מושלמת". אם כן – הפעולה תחזיר ערך true, אם לא – הפעולה תחזיר ערך false.

כותרת הפעולה :

```
public static bool PerfectList(Node<int> chain)
```

(2 נק') ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

שאלה 4

נתונות שתי מחלקות A ו-B:

```

public class A
{
    protected int num;
    public A() { }
    public A (int num){ this.num = num; }
    public int Func(int k){ return k + this.num;}
}
public class B : A {
    private int x;
    public bool Test() { return this.x > this.num; }
}

```

(6 נק') א. לפניכם שלושה היגדים. קבעו לכל אחד מהם אם הוא נכון או אינו נכון, ונמקו את קביעתכם.

1. המחלקה A לא תעבור קומפילציה כי התכונה num אינה מקבלת ערך בבנאי הראשון.
2. המחלקה B לא תעבור קומפילציה כי אין לה שום בנאי.
3. המחלקה B לא תעבור קומפילציה כי בפעולה test יש שימוש בתכונה num והיא לא הוגדרה במחלקה זו.

(5 נק') ב. לכל אחת מהפעולות הבאות קבעו אם אפשר להגדיר אותה במחלקה A. אם הוספת הפעולה תגרום לשגיאה יש להסביר אותה.

1. `public int Func () { return 1; }`
2. `public int Func (double k){ return (int) k; }`
3. `public double Func (int k){ return 1.0*k; }`
4. `public int Func (int p){ return p; }`
5. `public void Func (int k, int p){ Console.WriteLine(k*p); }`

(5 נק') ג. כותרת הפעולה Func במחלקה A השתנתה:

```
public virtual int Func(int k)
```

לכל אחת מהפעולות הבאות קבעו אם אפשר להגדיר אותה במחלקה B. אם הוספת הפעולה תגרום לשגיאה יש להסביר אותה

1. `public overload int Func () { return 1; }`
2. `public overload int Func (double k){ return (int) k; }`
3. `public overload double Func (int k){ return 1.0*k; }`
4. `public overload int Func (int p){ return p; }`
5. `public overload void Func (int k, int p){
Console.WriteLine(k*p); }`

שאלה 5

נתונה הפעולה הבאה:

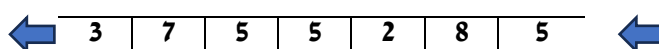
```

public static void What(Queue<int> q)
{
    int num1, num2;
    bool found;
    Queue<int> temp1 = new Queue<int>();
    Queue<int> temp2 = new Queue<int>();
    while(!q.IsEmpty())
    {
        num1 = q.Remove();
        found = false;
        while(!q.IsEmpty())
        {
            num2 = q.Remove();
            if(num1 == num2)
                found = true;
            temp1.Insert(num2);
        }
        while(!temp1.IsEmpty())
            q.Insert(temp1.Remove());

        if(!found)
            temp2.Insert(num1);
    }
    while(!temp2.IsEmpty())
        q.Insert(temp2.Remove());
}

```

(7 נק') א. נתון תור q הבא:



מה יהיה התוכן של התור q אחרי זימון הפעולה What(q) עבור התור q?

יש להראות מעקב אחרי ביצוע הפעולה What.

(3 נק') ב. תנו דוגמה לתור q המכיל לפחות חמישה ערכים כך שתוכן התור לא ישתנה אחרי זימון הפעולה What(q). אם לא ניתן ליצור תור כזה, יש להסביר מדוע.

(3 נק') ג. תנו דוגמה לתור q המכיל לפחות חמישה ערכים כך שאחרי זימון הפעולה What(q) התור יהיה ריק. אם לא ניתן ליצור תור כזה, יש להסביר מדוע.

(3 נק') ד. מה מבצעת הפעולה What(q) עבור תור של מספרים שלמים q?

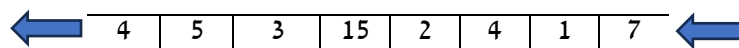
חלק ב'

ענו על שתיים מבין השאלות 6–10 (ערך כל שאלה – 18 נקודות).

שאלה 6

הגדרה 1: תור "מושלם N" הוא תור של מספרים שלמים המכיל את כל המספרים שבין 1 ל-N (כולל).
לדוגמה:

התור הבא הוא תור מושלם של 5 אבל לא תור מושלם של 7 (כי מספר 6 אינו מופיע בתור).



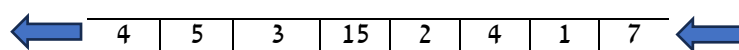
(8 נק') א.

כתבו פעולה `bool IsPerfectN(Queue<int> q, int n)` המקבלת תור של מספרים שלמים `q` ומספר שלם `n` וחיובי `n`. הפעולה תבדוק אם תור הוא "תור מושלם של `n`". אם כן – הפעולה תחזיר `true`, ואם לא – הפעולה תחזיר `false`.

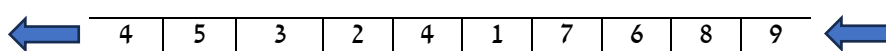
הגדרה 2: תור "סופר מושלם N" הוא תור של מספרים שלמים המכיל את כל המספרים שבין 1 ל-N (כולל) ורק אותם.

(8 נק') ב.

כתבו פעולה `void DoItSuper(Queue<int> q, int n)` המקבלת תור של מספרים שלמים `q` ומספר שלם `n` וחיובי `n`. הפעולה מורידה מתור ערכים מיותרים ומוסיפה בסוף התור ערכים חסרים, כך שיתקבל תור "סופר מושלם `n`".
אם לפני זימון הפעולה התור `q` היה



אז אחרי ביצוע זימון הפעולה `DoItSuper(q, 9)` התור `q` יהיה:



(2 נק') ג. מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'–ב'? הסבירו את תשובתכם.

שאלה 7

(8 נק') א. כתבו פעולה המקבלת מחסנית מספרים שלמים `s` ומספר שלם `num`. הפעולה תבדוק אם קיים במחסנית זוג איברים שהסכום שלהם שווה ל-`num`. אם כן – הפעולה תחזיר `true`, ואם לא – הפעולה תחזיר `false`.

כותרת הפעולה:

```
public static bool ExistSum(Stack<int> s, int num)
```

(8 נק') ב. כתבו פעולה המקבלת מחסנית של מספרים שלמים `s` ומחזירה את הסכום הגדול ביותר של שני איברי המחסנית. אפשר להניח שבמחסנית יש לפחות שני איברים.

כותרת הפעולה:

```
public static int MaxSum(Stack<int> s)
```

(2 נק') ג. מהן סיבוכיות הפעולות שכתבתם בסעיפים א' ו-ב'? הסבירו את תשובתכם.

שאלה 8**הגדרה:**

קבוצת מספרים – `SetOfNumbers` – היא אוסף של מספרים, כאשר כל מספר יכול להופיע בקבוצה פעם אחת בלבד. אין חשיבות לסדר המספרים בקבוצה.

לפניכם ממשק חלקי של המחלקה `SetOfNumbers` המייצגת קבוצת מספרים שלמים.

כותרת הפעולה	תיאור הפעולה
<code>public SetOfNumbers ()</code>	הפעולה בונה קבוצת מספרים ריקה
<code>public void AddToSet(int num)</code>	הפעולה מוסיפה לקבוצה את המספר <code>num</code> , אם המספר לא נמצא בקבוצה. אם המספר <code>num</code> כבר נמצא בקבוצה, הפעולה לא עושה דבר.
<code>public int RemoveRandom()</code>	הפעולה מחזירה באופן אקראי אחד מאיברי הקבוצה ומוחקת אותו מהקבוצה. הנחה: הקבוצה לא ריקה.
<code>public bool IsEmpty()</code>	הפעולה מחזירה <code>true</code> אם הקבוצה הנוכחית ריקה, אחרת – הפעולה מחזירה <code>false</code>

סיבוכיות של כל הפעולות האלה $O(1)$

שימו לב! מימוש של המחלקה `SetOfNumbers` לא ידוע, אפשר להשתמש אך ורק בפעולות הנתונות בממשק!

(5 נק') א. כתבו במחלקה `SetOfNumbers` פעולה פנימית המחזירה את מספר האיברים בקבוצה.

יש לשמור על הקבוצה אחרי ביצוע הפעולה.

כותרת הפעולה:

```
public int SizeOfSet()
```

(5 נק') ב. כתבו במחלקה `SetOfNumbers` פעולה פנימית המחזירה את המספר הקטן ביותר בקבוצה

ומוחקת אותו מהקבוצה.

הנחה: הקבוצה לא ריקה.

כותרת הפעולה:

```
public int RemoveMin()
```

(5 נק') ג. כתבו פעולה חיצונית המקבלת שתי קבוצות `sn1` ו-`sn2`. הפעולה תחזיר `true` אם כל

האיברים בקבוצה `sn1` גדולים מכל האיברים בקבוצה `sn2`, אם לא – הפעולה תחזיר `false`.

יש לשמור על הקבוצה אחרי ביצוע הפעולה.

הנחה: הקבוצות לא ריקות.

כותרת הפעולה:

```
public static bool Bigger(SetOfNumbers sn1, SetOfNumbers sn2)
```

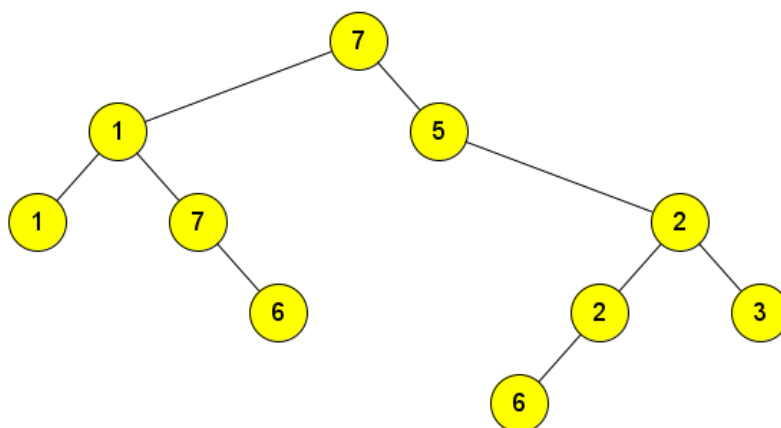
(3 נק') ד. מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'–ג'? הסבירו את תשובתכם.

שאלה 9

נתונה הפעולה `what`. הפעולה מקבלת הפניה לשורש עץ בינרי של מספרים שלמים חיוביים `bt` ושני מספרים שלמים חיוביים `num` ו-`count`.

```
public static bool What(BinNode<int> bt,
                        int num, int count)
{
    if(bt == null) return false;
    if(num < 1 || count < 1) return false;
    if(bt.GetLeft() == null && bt.GetRight() == null)
    {
        return bt.GetValue() == num && count == 1;
    }
    return What(bt.GetLeft(), num - bt.GetValue(), count - 1) ||
           What(bt.GetRight(), num - bt.GetValue(), count - 1);
}
```

נתון העץ הבינרי הבא:



6 נק') א. מה תהיה תוצאת זימון הפעולה `What(bt, 16, 4)`?

יש להראות מעקב אחרי ביצוע הפעולה `What`.

3 נק') ב. תנו דוגמה לערכים של `num` ו-`count` כך שהזימון `What(bt, num, count)` יחזיר תוצאה שונה מזו שהתקבלה בסעיף א'.

3 נק') ג. האם קיים ערך `count > 0` שעבורו הזימון `What(bt, num, count)` יחזיר ערך `false` לכל `num > 0`? הסבירו את תשובתכם.

3 נק') ד. תנו דוגמה לעץ בינרי של מספרים שלמים חיוביים `bt` שעבורו כל אחד משלושת זימונים הבאים יחזיר ערך `true`:

`What(bt, 2, 2)`

`What(bt, 3, 3)`

`What(bt, 4, 4)`

3 נק') ה. מה מבצעת הפעולה `What` באופן כללי?

שאלה 10

בבית האבות "גיל הזהב" הוחלט לצרף לכל דייר איש קשר שדרכו משפחות הדיירים יוכלו לקבל מידע עדכני על מצבו הבריאותי של הדייר ועל תוכניות הבידור והאירועים בבית האבות. איש הקשר הוא דייר בבית האבות שהתנדב לתפקיד זה. לשם כך פותחו שלוש המחלקות הבאות: `Test`, `Contact`, `Person`. המחלקה `Person` מתארת דייר:

```
public class Person {
    private string fName; // שם פרטי
    private string lName; // שם משפחה
    private int age; // גיל
    public Person() {
        this.fName = "unknown";
        this.lName = "unknown";
        this.age = 120;
    }
    public Person(string fName, string lName, int age) {
        this.fName = fName;
        this.lName = lName;
        this.age = age;
    }
    public Person(Person other) {
        this.fName = other.fName;
        this.lName = other.lName;
        this.age = other.age;
    }
    public void AddAge(int num) {
        if(this.age + num < 120)
            this.age += num;
    }
    public override string ToString() {
        return this.fName+" "+this.lName + " "+this.age;
    }
}
```

המחלקה `Contact` מתארת איש קשר:

```
public class Contact : Person {
    private string tel; // טלפון של איש קשר
    private Person contact; // איש קשר
    public Contact(string fName, string lName, int age):
        base(fName, lName, age) {
        this.tel = "NO";
        this.contact = new Person();
    }
    public Contact(Person person): base(person) {
        this.tel = "NO";
        this.contact = null;
    }
    public Contact(Person person1, Person person2, string tel):
        base(person1) {
        this.tel = tel;
        this.contact = new Person(person2);
    }
}
```

```
public Contact(string fName, string lName, int age,
               Person person, String tel):
    base(fName, lName, age) {
    this.tel = tel;
    this.contact = person;
}
public void SetContact(Person person, string tel){
    this.contact = person;
    this.tel = tel;
}
public override string ToString(){
    if(this.contact == null) return base.ToString();
    return base.ToString() + " --> " + this.contact +
        " tel: " + this.tel;
}
}
```

(12 נק') א. נתונה הפעולה הראשית main אשר נמצאת במחלקה Test. עקבו אחרי הביצוע של הפעולה main ורשמו מה יהיו ערכי התכונות של כל עצם שנוצר במהלך הביצוע.

(6 נק') ב. מהו הפלט של הפעולה ?Main

```
public static void Main(string[] args){
    Person[] p = new Person[6];
    p[0] = new Person("Avi", "Barak", 60);
    p[1] = new Person("Benny", "Coehn", 70);
    p[2] = new Person(p[1]);
    p[3] = p[1];
    p[4] = new Person();
    p[5] = new Person("David", "Ezra", 65);
    p[3].AddAge(30);
    for(int i = 0; i < p.Length; i++)
        Console.WriteLine(i + " " + p[i]);

    Person[] c = new Person[6];
    c[0] = new Contact(p[0], p[5], "123");
    c[1] = new Contact(p[1]);
    c[2] = new Contact(p[5], p[2], "456");
    c[3] = new Contact("Eric", "Far", 65, c[0], "789");
    c[4] = new Contact("Fiona", "Gold", 40);
    c[5] = (Person)c[3];
    ((Contact)c[1]).SetContact(c[5], "999");
    for(int i = 0; i < L.length; i++)
        Console.WriteLine(i + " " + c[i]);
}
```

חלק ג'

ענו על אחת מבין השאלות 11–12 (ערך שאלה – 16 נקודות).

שאלה 11

המרכז הרפואי "כל הבריאות" מפעיל שירות של אישור תורים אוטומטי. המערכת מזכירה לכל אדם שקבע תור למכון כלשהו את פרטי התור ומקבלת ממנו אישור הגעה או ביטול התור. פרטי התור כוללים את שם המכון, תאריך התור, שעת התור. לכל לקוח מתקשרים פעם אחת בשבוע. על הלקוח לאשר את הגעתו באמצעות הקשה על הספרה 1, או לבטל את התור באמצעות הקשה על הספרה 2. אם אין מענה, הלקוח עובר לרשימת התזכורת של שבוע הבא. המערכת מחדשת את פרטי תזכורות התורים מדי שבוע.

המחלקה **Reminder** מייצגת תזכורת:

תכונות המחלקה:

```
private string cust; // שם הלקוח
private string tel; // טלפון הלקוח
private string inst; // שם המכון
private string date; // תאריך התור
private int hour; // שעת התור
private int status; // 0- אין מענה, 1- אישור, 2- ביטול,
במחלקה הוגדרו פעולה בונה המקבלת את השם ומספר הטלפון של הלקוח, שם המכון, התאריך והשעה של התור. סטטוס התזכורת מאוחל ב-0 (אין מענה). במחלקה הוגדרו הפעולות Set/Get לכל התכונות. המחלקה DailyReminder מייצגת רשימת תזכורות ביום מסוים.
```

(2 נק') א. כתבו את כותרת המחלקה **DailyReminder** ואת התכונות שלה.

לייצוג אוסף (רשימה) אפשר להשתמש במחלקות **Node**, **Queue**, **Stack**. אפשר להוסיף

תכונות נוספות. חובה לתעד את התכונות.

המחלקה **WeeklyReminder** מייצגת מערכת שבועית של תזכורות. במחלקה יש תכונה אחת – מערך של תזכורות בכל יום בשבוע (1–6). תא ה-0 מיועד לתזכורות שלא קיבלו מענה.

```
private DailyReminder [] arr = new DailyReminder[7];
```

(2 נק') ב. כתבו במחלקה **WeeklyReminder** פעולה המקבלת את הפרטים של הלקוח ושל התור ואת

יום התזכורת **dayReminder** (1–6) ומוסיפה את התזכורת לרשימת התזכורות המתאימה.

כותרת הפעולה:

```
public void AddReminder(string cust, string tel, string inst,
                        string date, int hour, int dayReminder)
```

(6 נק') ג. כתבו במחלקה **WeeklyReminder** פעולה המקבלת את שם הלקוח **cust**, שם המכון **inst**,

יום התזכורת **dayReminder** (1–6) וקוד אישור ההגעה **answer** (0/1/2).

אם ערך קוד האישור הוא 1 או 2, הפעולה תעדכן את הסטטוס בהתאם. אם אין מענה, הפעולה

תעביר את התזכורת לרשימת תזכורות בתא 0 כדי לטפל בה בשבוע הבא. כותרת הפעולה:

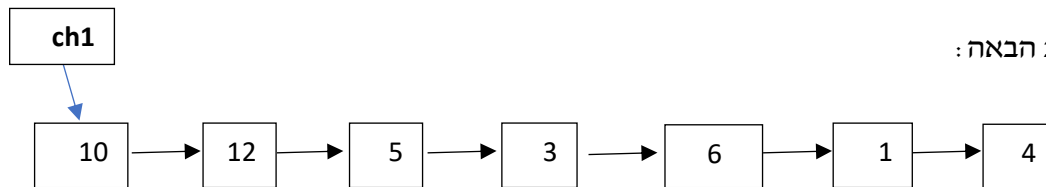
```
public void UpdateReminder(string cust, string inst,
                           int dayReminder, int answer)
```

(6 נק') ד. כתבו במחלקה **WeeklyReminder** פעולה המקבלת מערך מכונים של המרכז הרפואי "כל

הבריאות". הפעולה תדפיס עבור כל מכון את פרטי התורים שהתפנו עקב ביטול מצד המטופל.

שאלה 12

נתונה השרשרת הבאה:



נתונות שתי הפעולות הרקורסיביות הבאות:

```

public static int Zero(Node<int> ch){
    if(ch == null) return 0;
    return 1 + Zero(ch.GetNext());
}

public static Node<int> One(Node<int> ch, int n){
    if(ch == null) return null;
    if(n == 1) return ch;
    return One(ch.GetNext(), n-1);
}
  
```

(2 נק') א. מה יהיה הפלט של הפקודה הבאה?

```
Console.WriteLine(Zero(ch1));
```

מה מבצעת הפעולה Zero?

(2 נק') ב. מה יהיה הפלט של הפקודה הבאה?

```
Console.WriteLine(One(ch1, 3));
```

מה מבצעת הפעולה one עבור ערך n שלם? יש להתייחס לאפשרויות שונות.

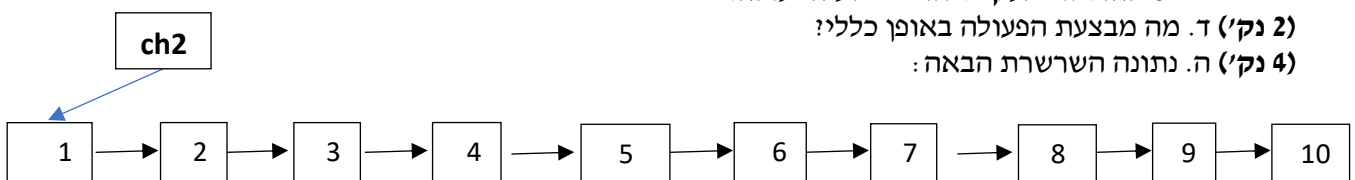
(6 נק') ג. נתונה הפעולה הרקורסיבית two הבאה:

```

public static void Two(Node<int> ch, int n, int m){
    if(n < m)
    {
        Node<int> p1 = One(ch, n);
        Node<int> p2 = One(ch, m);
        int temp = p1.GetValue();
        p1.SetValue(p2.GetValue());
        p2.SetValue(temp);
        Two(ch, n+1, m-1);
    }
}
  
```

עקבו אחרי זימון הפעולה Two(ch1, 2, 6) ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה?

יש להראות מעקב אחרי ביצוע הפעולה.

(2 נק') ד. מה מבצעת הפעולה באופן כללי?**(4 נק')** ה. נתונה השרשרת הבאה:

איך תיראה השרשרת לאחר ביצוע קטע קוד הבא:

```

int s = Zero(ch2);
for(int i = 1; i <= s/2; i++){
    Two(ch2, i, s - i + 1);
}
  
```

מבחן ב-JAVA**חלק א'**

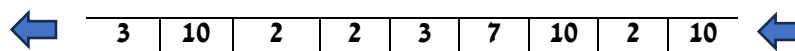
ענו על שלוש מבין השאלות 1–5 (ערך כל שאלה – 16 נקודות).

שאלה 1

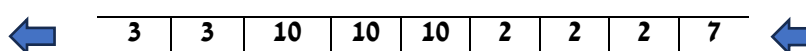
(14 נק') א. כתבו פעולה המקבלת תור של מספרים שלמים ומחזירה תור חדש המכיל את כל האיברים שבתור המקורי כך שהאיברים הזהים יהיו ברצף. סדר הרצפים בתור החדש אינו חשוב.

לדוגמה ,

עבור התור q הזה:



הפעולה תחזיר את התור (אחת התשובות האפשריות):



כותרת הפעולה:

```
public static Queue<Integer> orderQ(Queue<Integer> q)
```

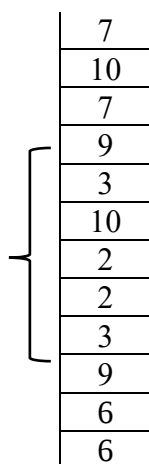
(2 נק') ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

שאלה 2

(14 נק') א. נתונה מחסנית של מספרים שלמים. כל איבר במחסנית מופיע פעמיים בדיוק. המרחק בין שני איברים במחסנית הוא כמות האיברים הנמצאים ביניהם.

כתבו פעולה המחזירה את המרחק הגדול ביותר בין שני איברים זהים במחסנית.

לדוגמה:



עבור המחסנית הבאה הפעולה תחזיר 5 (המרחק בין שני איברים שערכם 9).

כותרת הפעולה:

```
public static int maxDistance(Stack<Integer> st)
```

(2 נק') ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

שאלה 3

(14 נק') א. שרשרת נקראת "שרשרת מושלמת" אם כל ערך חיובי שבה מופיע מספר זוגי של פעמים וכל ערך שלילי שבה מופיע מספר אי-זוגי של פעמים.

כתבו פעולה המקבלת הפניה לחוליה הראשונה בשרשרת של מספרים שלמים. הפעולה תבדוק אם השרשרת היא "שרשרת מושלמת". אם כן – הפעולה תחזיר ערך true, אם לא – הפעולה תחזיר ערך false.

כותרת הפעולה:

```
public static boolean perfectList(Node<Integer> chain)
```

(2 נק') ב. מהי סיבוכיות הפעולה שכתבתם בסעיף א'? הסבירו את תשובתכם.

שאלה 4

נתונות שתי מחלקות A ו-B:

```

public class A
{
    protected int num;
    public A() { }
    public A (int num){ this.num = num; }
    public int func(int k){ return k + this.num;}
}
public class B extends A {
    private int x;
    public boolean test() { return this.x > this.num; }
}

```

(6 נק') א. לפניכם שלושה היגדים. קבעו לכל אחד מהם אם הוא נכון או אינו נכון, ונמקו את קביעתכם.

1. המחלקה A לא תעבור קומפילציה כי התכונה num אינה מקבלת ערך בבנאי הראשון.
2. המחלקה B לא תעבור קומפילציה כי אין לה שום בנאי.
3. המחלקה B לא תעבור קומפילציה כי בפעולה test יש שימוש בתכונה num והיא לא הוגדרה במחלקה זו.

(5 נק') ב. לכל אחת מהפעולות הבאות קבעו אם אפשר להגדיר אותה במחלקה A. אם הוספת הפעולה תגרום לשגיאה יש להסביר אותה.

1. `public int func () { return 1; }`
2. `public int func (double k){ return (int) k; }`
3. `public double func (int k){ return 1.0*k; }`
4. `public int func (int p){ return p; }`
5. `public void func(int k, int p){ System.out.print(k*p); }`

(5 נק') ג. לכל אחת מהפעולות הבאות קבעו אם אפשר להגדיר אותה במחלקה B. אם הוספת הפעולה תגרום לשגיאה יש להסביר אותה.

1. `public int func () { return 1; }`
2. `public int func (double k){ return (int) k; }`
3. `public double func (int k){ return 1.0*k; }`
4. `public int func (int p){ return p; }`
5. `public void func(int k, int p){ System.out.print(k*p); }`

שאלה 5

נתונה הפעולה הבאה:

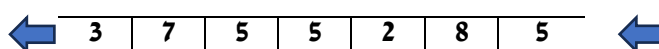
```

public static void what(Queue<Integer> q)
{
    int num1, num2;
    boolean found;
    Queue<Integer> temp1 = new Queue<Integer>();
    Queue<Integer> temp2 = new Queue<Integer>();
    while(!q.isEmpty())
    {
        num1 = q.remove();
        found = false;
        while(!q.isEmpty())
        {
            num2 = q.remove();
            if(num1 == num2)
                found = true;
            temp1.insert(num2);
        }
        while(!temp1.isEmpty())
            q.insert(temp1.remove());

        if(!found)
            temp2.insert(num1);
    }
    while(!temp2.isEmpty())
        q.insert(temp2.remove());
}

```

(7 נק') א. נתון תור q הבא:



מה יהיה התוכן של התור q אחרי זימון הפעולה what (q) עבור התור q?
יש להראות מעקב אחרי ביצוע הפעולה what.

(3 נק') ב. תנו דוגמה לתור q המכיל לפחות חמישה ערכים כך שתוכן התור לא ישתנה אחרי זימון הפעולה what (q). אם לא ניתן ליצור תור כזה, יש להסביר מדוע.

(3 נק') ג. תנו דוגמה לתור q המכיל לפחות חמישה ערכים כך שאחרי זימון הפעולה what (q) התור יהיה ריק. אם לא ניתן ליצור תור כזה, יש להסביר מדוע.

(3 נק') ד. מה מבצעת הפעולה what (q) עבור תור של מספרים שלמים q?

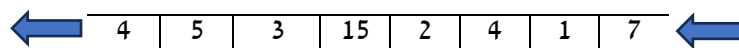
חלק ב'

ענו על שתיים מבין השאלות 6–10 (ערך כל שאלה – 18 נקודות).

שאלה 6

הגדרה 1: תור "מושלם N" הוא תור של מספרים שלמים המכיל את כל המספרים שבין 1 ל-N (כולל). לדוגמה:

התור הבא הוא תור מושלם של 5 אבל לא תור מושלם של 7 (כי מספר 6 אינו מופיע בתור).



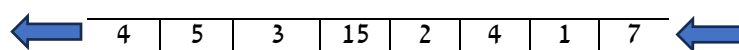
(8 נק') א.

כתבו פעולה `boolean isPerfectN(Queue<Integer> q, int n)` המקבלת תור של מספרים שלמים `q` ומספר שלם וחיובי `n`. הפעולה תבדוק אם תור הוא "תור מושלם של n". אם כן – הפעולה תחזיר ערך `true`, ואם לא – הפעולה תחזיר ערך `false`.

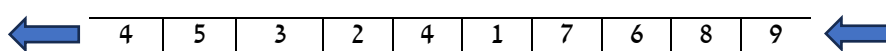
הגדרה 2: תור "סופר מושלם N" הוא תור של מספרים שלמים המכיל את כל המספרים שבין 1 ל-N (כולל) ורק אותם.

(8 נק') ב.

כתבו פעולה `void doItSuper(Queue<Integer> q, int n)` המקבלת תור של מספרים שלמים `q` ומספר שלם וחיובי `n`. הפעולה מורידה מתור ערכים מיותרים ומוסיפה בסוף התור ערכים חסרים, כך שיתקבל תור "סופר מושלם n". אם לפני זימון הפעולה התור `q` היה



אז אחרי ביצוע זימון הפעולה `doItSuper(q, 9)` התור `q` יהיה:



(2 נק') ג. מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'–ב'? הסבירו את תשובתכם.

שאלה 7

(8 נק') א. כתבו פעולה המקבלת מחסנית מספרים שלמים `s` ומספר שלם `num`. הפעולה תבדוק אם קיים במחסנית זוג איברים שהסכום שלהם שווה ל-`num`. אם כן – הפעולה תחזיר ערך `true`, ואם לא – הפעולה תחזיר ערך `false`.

כותרת הפעולה:

```
public static boolean existSum(Stack<Integer> s, int num)
```

(8 נק') ב. כתבו פעולה המקבלת מחסנית של מספרים שלמים `s` ומחזירה את הסכום הגדול ביותר של שני איברי המחסנית. אפשר להניח שבמחסנית יש לפחות שני איברים.

כותרת הפעולה:

```
public static int maxSum(Stack<Integer> s)
```

(2 נק') ג. מהן סיבוכיות הפעולות שכתבתם בסעיפים א' ו-ב'? הסבירו את תשובתכם.

שאלה 8**הגדרה:**

קבוצת מספרים – `SetOfNumbers` – היא אוסף של מספרים, כאשר כל מספר יכול להופיע בקבוצה פעם אחת בלבד. אין חשיבות לסדר המספרים בקבוצה.

לפניכם ממשק חלקי של המחלקה `SetOfNumbers` המייצגת קבוצת מספרים שלמים.

תיאור הפעולה	כותרת הפעולה
הפעולה בונה קבוצת מספרים ריקה	<code>public SetOfNumbers ()</code>
הפעולה מוסיפה לקבוצה את המספר <code>num</code> , אם המספר לא נמצא בקבוצה. אם המספר <code>num</code> כבר נמצא בקבוצה, הפעולה לא עושה דבר.	<code>public void addToSet(int num)</code>
הפעולה מחזירה באופן אקראי אחד מאיברי הקבוצה ומוחקת אותו מהקבוצה. הנחה: הקבוצה לא ריקה.	<code>public int removeRandom()</code>
הפעולה מחזירה <code>true</code> אם הקבוצה הנוכחית ריקה, אחרת – הפעולה מחזירה <code>false</code>	<code>public boolean isEmpty()</code>

סיבוכיות של כל הפעולות האלה $O(1)$

שימו לב! מימוש של המחלקה `SetOfNumbers` לא ידוע, אפשר להשתמש אך ורק בפעולות הנתונות בממשק!

(5 נק') א. כתבו במחלקה `SetOfNumbers` פעולה פנימית המחזירה את מספר האיברים בקבוצה.

יש לשמור על הקבוצה אחרי ביצוע הפעולה.

כותרת הפעולה:

```
public int sizeOfSet()
```

(5 נק') ב. כתבו במחלקה `SetOfNumbers` פעולה פנימית המחזירה את המספר הקטן ביותר בקבוצה

ומוחקת אותו מהקבוצה.

הנחה: הקבוצה לא ריקה.

כותרת הפעולה:

```
public int removeMin()
```

(5 נק') ג. כתבו פעולה חיצונית המקבלת שתי קבוצות `sn1` ו-`sn2`. הפעולה תחזיר `true` אם כל

האיברים בקבוצה `sn1` גדולים מכל האיברים בקבוצה `sn2`, אם לא – הפעולה תחזיר `false`.

יש לשמור על הקבוצה אחרי ביצוע הפעולה.

הנחה: הקבוצות לא ריקות.

כותרת הפעולה:

```
public static boolean bigger(SetOfNumbers sn1, SetOfNumbers sn2)
```

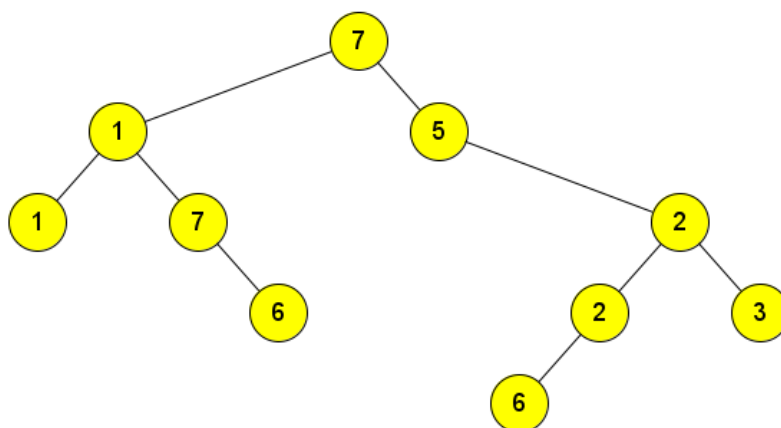
(3 נק') ד. מהן הסיבוכיות של הפעולות שכתבתם בסעיפים א'–ג'? הסבירו את תשובתכם.

שאלה 9

נתונה הפעולה `what`. הפעולה מקבלת הפניה לשורש עץ בינרי של מספרים שלמים חיוביים `bt` ושני מספרים שלמים חיוביים `num` ו-`count`.

```
public static boolean what(BinNode<Integer> bt,
                           int num, int count)
{
    if(bt == null) return false;
    if(num < 1 || count < 1) return false;
    if(bt.getLeft() == null && bt.getRight() == null)
    {
        return bt.getValue() == num && count == 1;
    }
    return what(bt.getLeft(), num - bt.getValue(), count - 1) ||
           what(bt.getRight(), num - bt.getValue(), count - 1);
}
```

נתון העץ הבינרי הבא:



6 נק') א. מה תהיה תוצאת זימון הפעולה `what(bt, 16, 4)`?

יש להראות מעקב אחרי ביצוע הפעולה `what`.

3 נק') ב. תנו דוגמה לערכים של `num` ו-`count` כך שהזימון `what(bt, num, count)` יחזיר תוצאה שונה מזו שהתקבלה בסעיף א'.

3 נק') ג. האם קיים ערך `count > 0` שעבורו הזימון `what(bt, num, count)` יחזיר ערך `false` לכל `num > 0`? הסבירו את תשובתכם.

3 נק') ד. תנו דוגמה לעץ בינרי של מספרים שלמים חיוביים `bt` שעבורו כל אחד משלושת זימונים הבאים יחזיר ערך `true`:

`what(bt, 2, 2)`

`what(bt, 3, 3)`

`what(bt, 4, 4)`

3 נק') ה. מה מבצעת הפעולה `what` באופן כללי?

שאלה 10

בבית האבות "גיל הזהב" הוחלט לצרף לכל דייר איש קשר שדרכו משפחות הדיירים יוכלו לקבל מידע עדכני על מצבו הבריאותי של הדייר ועל תוכניות הבידור והאירועים בבית האבות. איש הקשר הוא דייר בבית האבות שהתנדב לתפקיד זה. לשם כך פותחו שלוש המחלקות הבאות: `Test`, `Contact`, `Person`. המחלקה `Person` מתארת דייר:

```
public class Person {
    private String fName; // שם פרטי
    private String lName; // שם משפחה
    private int age; // גיל
    public Person() {
        this.fName = "unknown";
        this.lName = "unknown";
        this.age = 120;
    }
    public Person(String fName, String lName, int age) {
        this.fName = fName;
        this.lName = lName;
        this.age = age;
    }
    public Person(Person other) {
        this.fName = other.fName;
        this.lName = other.lName;
        this.age = other.age;
    }
    public void addAge(int num) {
        if(this.age + num < 120)
            this.age += num;
    }
    public String toString(){
        return this.fName+" "+this.lName + " "+this.age;
    }
}
```

המחלקה `Contact` מתארת איש קשר:

```
public class Contact extends Person {
    private String tel; // טלפון של איש קשר
    private Person contact; // איש קשר
    public Contact(String fName, String lName, int age){
        super(fName,lName,age);
        this.tel = "NO";
        this.contact = new Person();
    }
    public Contact(Person person){
        super(person);
        this.tel = "NO";
        this.contact = null;
    }
    public Contact(Person person1, Person person2, String tel){
        super(person1);
        this.tel = tel;
        this.contact = new Person(person2);
    }
}
```

```
public Contact(String fName, String lName, int age,
                Person person, String tel){
    super(fName, lName, age);
    this.tel = tel;
    this.contact = person;
}
public void setContact(Person person, String tel){
    this.contact = person;
    this.tel = tel;
}
public String toString(){
    if(this.contact == null) return super.toString();
    return super.toString() + " --> " + this.contact +
        " tel: " + this.tel;
}
}
```

(12 נק') א. נתונה הפעולה הראשית main אשר נמצאת במחלקה Test. עקבו אחרי הביצוע של הפעולה main ורשמו מה יהיו ערכי התכונות של כל עצם שנוצר במהלך הביצוע.

(6 נק') ב. מהו הפלט של הפעולה ?main

```
public static void main(String[] args){
    Person[] p = new Person[6];
    p[0] = new Person("Avi", "Barak", 60);
    p[1] = new Person("Benny", "Coehn", 70);
    p[2] = new Person(p[1]);
    p[3] = p[1];
    p[4] = new Person();
    p[5] = new Person("David", "Ezra", 65);
    p[3].addAge(30);
    for(int i = 0; i < p.length; i++)
        System.out.println(i + " " + p[i]);

    Person[] c = new Person[6];
    c[0] = new Contact(p[0], p[5], "123");
    c[1] = new Contact(p[1]);
    c[2] = new Contact(p[5], p[2], "456");
    c[3] = new Contact("Eric", "Far", 65, c[0], "789");
    c[4] = new Contact("Fiona", "Gold", 40);
    c[5] = (Person) c[3];
    ((Contact) c[1]).setContact(c[5], "999");
    for(int i = 0; i < c.length; i++)
        System.out.println(i + " " + c[i]);
}
```

חלק ג'

ענו על אחת מבין השאלות 11–12 (ערך שאלה – 16 נקודות).

שאלה 11

המרכז הרפואי "כל הבריאות" מפעיל שירות של אישור תורים אוטומטי. המערכת מזכירה לכל אדם שקבע תור למכון כלשהו את פרטי התור ומקבלת ממנו אישור הגעה או ביטול התור. פרטי התור כוללים את שם המכון, תאריך התור, שעת התור. לכל לקוח מתקשרים פעם אחת בשבוע. על הלקוח לאשר את הגעתו באמצעות הקשה על הספרה 1, או לבטל את התור באמצעות הקשה על הספרה 2. אם אין מענה, הלקוח עובר לרשימת התזכורת של שבוע הבא. המערכת מחדשת את פרטי תזכורות התורים מדי שבוע.

המחלקה **Reminder** מייצגת תזכורת:

תכונות המחלקה:

```
private String cust; // שם הלקוח
private String tel; // טלפון הלקוח
private String inst; // שם המכון
private String date; // תאריך התור
private int hour; // שעת התור
private int status; // 0- אין מענה, 1- אישור, 2- ביטול,
במחלקה הוגדרו פעולה בונה המקבלת את השם ומספר הטלפון של הלקוח, שם המכון, התאריך והשעה של התור. סטטוס התזכורת מאוחל ב-0 (אין מענה). במחלקה הוגדרו הפעולות set/get לכל התכונות. המחלקה DailyReminder מייצגת רשימת תזכורות ביום מסוים.
```

(2 נק') א. כתבו את כותרת המחלקה **DailyReminder** ואת התכונות שלה.

לייצוג אוסף (רשימה) אפשר להשתמש במחלקות **Node**, **Queue**, **Stack**. אפשר להוסיף

תכונות נוספות. חובה לתעד את התכונות.

המחלקה **WeeklyReminder** מייצגת מערכת שבועית של תזכורות. במחלקה יש תכונה אחת – מערך של תזכורות בכל יום בשבוע (1–6). תא ה-0 מיועד לתזכורות שלא קיבלו מענה.

```
private DailyReminder [] arr = new DailyReminder[7];
```

(2 נק') ב. כתבו במחלקה **WeeklyReminder** פעולה המקבלת את הפרטים של הלקוח ושל התור ואת

יום התזכורת **dayReminder** (1–6) ומוסיפה את התזכורת לרשימת התזכורות המתאימה.

כותרת הפעולה:

```
public void addReminder(String cust, String tel, String inst,
                        String date, int hour, int dayReminder)
```

(6 נק') ג. כתבו במחלקה **WeeklyReminder** פעולה המקבלת את שם הלקוח **cust**, שם המכון **inst**,

יום התזכורת **dayReminder** (1–6) וקוד אישור ההגעה **answer** (0/1/2).

אם ערך קוד האישור הוא 1 או 2, הפעולה תעדכן את הסטטוס בהתאם. אם אין מענה, הפעולה

תעביר את התזכורת לרשימת תזכורות בתא 0 כדי לטפל בה בשבוע הבא. כותרת הפעולה:

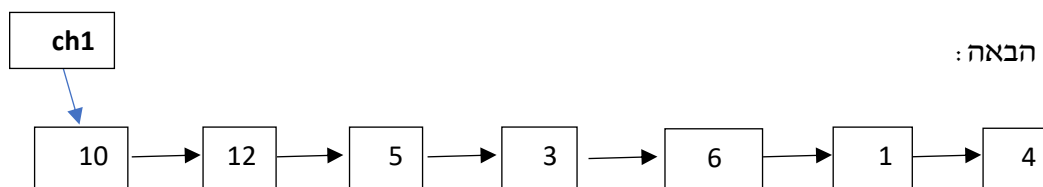
```
public void updateReminder(String cust, String inst,
                           int dayReminder, int answer)
```

(6 נק') ד. כתבו במחלקה **WeeklyReminder** פעולה המקבלת מערך מכונים של המרכז הרפואי "כל

הבריאות". הפעולה תדפיס עבור כל מכון את פרטי התורים שהתפנו עקב ביטול מצד המטופל.

שאלה 12

נתונה השרשרת הבאה:



נתונות שתי הפעולות הרקורסיביות הבאות:

```

public static int zero(Node<Integer> ch){
    if(ch == null) return 0;
    return 1 + zero(ch.getNext());
}

public static Node<Integer> one(Node<Integer> ch, int n){
    if(ch == null) return null;
    if(n == 1) return ch;
    return one(ch.getNext(), n-1);
}

```

(2 נק') א. מה יהיה הפלט של הפקודה הבאה?

```
System.out.println(zero(ch1));
```

מה מבצעת הפעולה zero?

(2 נק') ב. מה יהיה הפלט של הפקודה הבאה?

```
System.out.println(one(ch1, 3));
```

מה מבצעת הפעולה one עבור ערך n שלם? יש להתייחס לאפשרויות שונות.

(6 נק') ג. נתונה הפעולה הרקורסיבית two הבאה:

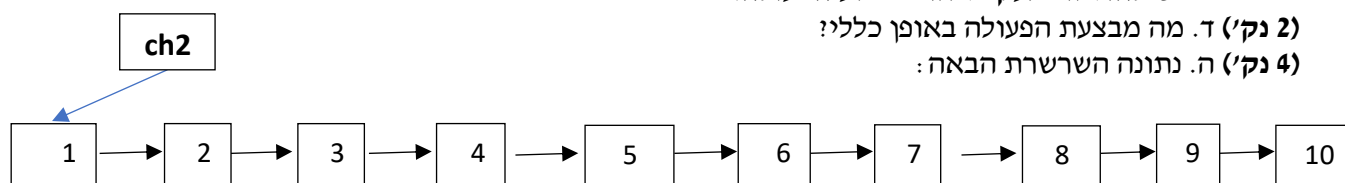
```

public static void two(Node<Integer> ch, int n, int m){
    if(n < m)
    {
        Node<Integer> p1 = one(ch, n);
        Node<Integer> p2 = one(ch, m);
        int temp = p1.getValue();
        p1.setValue(p2.getValue());
        p2.setValue(temp);
        two(ch, n+1, m-1);
    }
}

```

עקבו אחרי זימון הפעולה two(ch1, 2, 6) ורשמו איך תיראה השרשרת אחרי ביצוע הפעולה?

יש להראות מעקב אחרי ביצוע הפעולה.

(2 נק') ד. מה מבצעת הפעולה באופן כללי?**(4 נק')** ה. נתונה השרשרת הבאה:

איך תיראה השרשרת לאחר ביצוע קטע קוד הבא:

```

int s = zero(ch2);
for(int i = 1; i <= s/2; i++){
    two(ch2, i, s - i + 1);
}

```