

קרא בעיון את ההנחיות שלהלן:

- במבחן יש חמש שאלות. עליכם לענות על כולן.
- כל התכניות צריכות להיות מתועדות היטב. יש לכתוב תחילה **בקצרה** את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית. יש לבחור בשמות משמעותיים למשתנים, לשיטות ולקבועים שבתכנית. תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85% מהניקוד.
- **אפשר לתעד בעברית.** אין צורך בתיעוד API.
- יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה. **תכנית לא יעילה לא תקבל את מלוא הנקודות.**
- אם ברצונך להשתמש בתשובתך בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיק את השיטה או את המחלקה למחברת הבחינה. מספיק להפנות למקום הנכון, ובלבד שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').
- אין להשתמש במחלקות קיימות ב-Java, חוץ מאלו המפורטות בשאלות הבחינה.
- יש לשמור על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרוע מהציון.
- בכתיבת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה. **אין להשתמש במשתנים גלובליים!**

- **את התשובות לשאלות 3 – 5 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!**

חומר עזר המותר בשימוש הוא:

1. חוברת השקפים של הקורס

2. ספר הלימוד Java Software Solutions

אסור להשתמש במחשב מכל סוג שהוא!

חלק א – עליכם לענות על כל השאלות בחלק זה במחברת הבחינה

שאלה 1 - 25 נקודות

נתון מערך בגודל n , וערך שלם חיובי \max

ברצוננו לשבץ במערך n מספרים מתוך המספרים 1 עד \max , כך שהמספרים יהיו מסודרים במערך בסדר לא יורד (כלומר, סדר עולה, אך יכולים להיות גם מספרים שווים).

דוגמאות:

אם גודל המערך הוא $n = 3$ ו- $\max = 2$ (כלומר, המספרים האפשריים הם 1 ו-2), אז יש לנו 4 אפשרויות לסדר את המערך כך שהמספרים יהיו בסדר לא יורד:

1. $\{1, 1, 1\}$
2. $\{1, 1, 2\}$
3. $\{1, 2, 2\}$
4. $\{2, 2, 2\}$

אם גודל המערך הוא $n = 2$ ו- $\max = 3$ (כלומר, המספרים האפשריים הם 1, 2 ו-3), אז יש לנו 6 אפשרויות לסדר את המערך כך שהמספרים יהיו בסדר לא יורד:

1. $\{1, 1\}$
2. $\{1, 2\}$
3. $\{1, 3\}$
4. $\{2, 2\}$
5. $\{2, 3\}$
6. $\{3, 3\}$

עליכם לכתוב שיטה **רקורסיבית** המקבלת גודל של מערך n ומספר מקסימלי \max ומחזירה את מספר האפשרויות שיש למלא את המערך במספרים מהתחום 1 עד \max כך שיהיו מסודרים בסדר לא יורד.

חתימת השיטה היא:

```
public static int howManySorted(int n, int max)
```

השיטה שתכתבו צריכה להיות **רקורסיבית** ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.

אין צורך לדאוג ליעילות השיטה!

שימו לב, פתרון מתמטי (מציאת נוסחה שתמצא את התשובה) לא יתקבל! עליכם לכתוב שיטה **רקורסיבית** לפתרון הבעיה.

אל תשכחו לתעד את מה שכתבתם!

שאלה 2 - 25 נקודות

נתונות השיטות הסטטיות הבאות:

```
private static int f (int[]a, int low, int high)
{
    int res = 0;
    for (int i=low; i<=high; i++)
        res += a[i];
    return res;
}

public static boolean what (int[]a, int num)
{
    for (int i=0; i<a.length; i++)
    {
        for (int j=i; j<a.length; j++)
        {
            if (f(a, i, j) == num)
                return true;
        }
    }
    return false;
}
```

- א. מה מבצעת השיטה what בהינתן לה מערך a מלא במספרים שלמים (חיוביים בלבד)?
הסבירו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת. כלומר, כתבו מה המשמעות של הערך המוחזר מהשיטה what. (4 נק')
- ב. מה סיבוכיות הזמן והמקום הכוללת של השיטה what ? (2 נק')
- ג. כתבו את השיטה what כך שתבצע את מה שביצעה בסעיף א בסיבוכיות זמן ריצה קטנה יותר. (17 נק')

שימו לב:

השיטה שתכתבו צריכה להיות יעילה ככל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט

נקודות בלבד.

- ד. מה סיבוכיות זמן הריצה והמקום של השיטה שכתבתם בסעיף ג? הסבירו תשובתכם. (2 נק')

אל תשכחו לתעד את מה שכתבתם!

**חלק ב - את התשובות לשאלות 3-5 יש לכתוב על גבי השאלון.
לא נבדוק תשובות שייכתבו במקום אחר!**

שאלה 3 - 15 נקודות

נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

```
public class Node
{
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int number) {
        _number = number;
        _leftSon = null;
        _rightSon = null;
    }

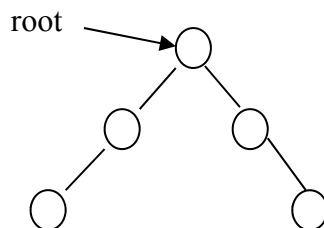
    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }
}
```

להלן נתונה השיטה הסטטית what המקבלת כפרמטר שורש של עץ בינרי.

```
public static int what (Node t)
{
    if (t==null)
        return -1;
    if ((t.getLeftSon()==null) && (t.getRightSon()==null))
        return 0;
    int n1 = what (t.getLeftSon());
    int n2 = what (t.getRightSon());
    if (n1 == n2)
        return n1+1;
    if (n1 > n2)
        return n2+1;
    else
        return n1+1;
}
```

סעיף א. (4%)

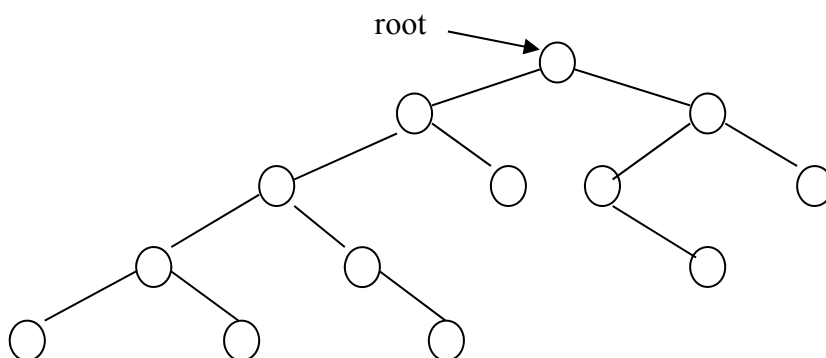
איזה ערך תחזיר השיטה what בהינתן לה השורש של העץ הבא :



התשובה היא :

סעיף ב. (4%)

איזה ערך תחזיר השיטה what בהינתן לה השורש של העץ הבא :



התשובה היא :

סעיף ג (7%)

מה מבצעת השיטה what בהינתן לה כפרמטר t מטיפוס Node שמצביע לשורש של עץ בינרי? כתבו בקצרה **מה** מבצעת השיטה ולא **כיצד** היא מבצעת זאת. כלומר, כתבו מה המשמעות של הערך שמוחזר מהשיטה.

שאלה 4 - 16 נקודות

נתונות המחלקות AAA, BBB, Driver הבאות (באותה חבילה):

```
public class AAA
{
    private int _val;

    public AAA()
    {
        _val=1;
    }

    public AAA(int val)
    {
        _val=val;
    }

    public int getVal()
    {
        return _val;
    }
}

//-----//

public class BBB extends AAA {
    private String _st;

    public BBB ()
    {
        _st="bbb";
    }

    public BBB(String st, int val)
    {
        super(val);
        _st=st;
    }

    public String getSt()
    {
        return _st;
    }
}
```

```

public boolean equals (Object other)
{
    System.out.print("Object ");
    if (other instanceof BBB)
        return (_st.equals(((BBB)other)._st) &&
            (getVal()==((BBB)other).getVal()));
    return false;
}

public boolean equals (AAA other)
{
    System.out.print("AAA ");
    if (other instanceof BBB)
        return (_st.equals(((BBB)other)._st) &&
            (getVal()==((BBB)other).getVal()));
    return false;
}

public boolean equals (BBB other)
{
    System.out.print("BBB ");
    return (_st.equals(other._st) &&
        (getVal()== other.getVal()));
}
}

public class Driver
{
    public static void main (String [] args)
    {
        AAA a1 = new AAA(5);
        AAA a2 = new AAA(5);
        AAA ab = new BBB();
        BBB b1 = new BBB();
        BBB b2 = new BBB();

        // שורות הקוד יתווספו פה
    }
}

```

נניח שאנחנו כותבים שורות קוד שונות בשיטה main של המחלקה Driver במקום ההערה:
שורות הקוד יתווספו פה // בכל סעיף כתבו מה יהיה הפלט (אם יהיה פלט).

שימו לב:

- בתוך השיטות equals השונות קיימות פקודות הדפסה, אל תשכחו להדפיס אותן.
- אין קשר בין שורות הקוד. בכל פעם אתם צריכים להניח שחמש שורות הקוד שכתובות כבר במחלקה קיימות, והשורה הנבדקת היא השורה השישית ב- main.

1. if (a1.equals(b1)) System.out.println(1);

יודפס

2. if (b1.equals(a1)) System.out.println(2);

יודפס

3. if (a1.equals(ab)) System.out.println(3);

יודפס

4. if (ab.equals(a1)) System.out.println(4);

יודפס

5. if (b1.equals(ab)) System.out.println(5);

יודפס

6. if (ab.equals(b1)) System.out.println(6);

יודפס

7. if (a1.equals(a2)) System.out.println(7);

יודפס

8. if (b1.equals(b2)) System.out.println(8);

יודפס

שאלה 5 - 19 נקודות

נתונה המחלקה IntNodeTwo הבאה, המייצגת איבר ברשימה מקושרת דו-סטריט בה יש מצביעים גם לאיבר הבא וגם לקודם, המכילה מספרים שלמים:

```
public class IntNodeTwo
{
    private int _num;
    private IntNodeTwo _next, _prev;

    public IntNodeTwo(int n) {
        _num = n;
        _next = null;
        _prev = null;
    }

    public IntNodeTwo(int num, IntNodeTwo n, IntNodeTwo p) {
        _num = num;
        _next = n;
        _prev = p;
    }

    public int getNum() { return _num; }
    public IntNodeTwo getNext() { return _next; }
    public IntNodeTwo getPrev() { return _prev; }
    public void setNum (int n) { _num = n; }
    public void setNext (IntNodeTwo node) { _next = node; }
    public void setPrev (IntNodeTwo node) { _prev = node; }
}
```

נתונה רשימה מקושרת דו-סטריט, המכילה מספרים שלמים הממומשת בעזרת המחלקה IntListTwo שלהלן:

```
public class IntListTwo
{
    private IntNodeTwo _head, _tail;
    public IntListTwo()
    {
        _head = null;
        _tail = null;
    }

    // כאן יש עוד בנאים ושיטות...
}
```

```

public int f() {
    int res = 0;
    IntNodeTwo temp = _head;
    while (temp != null)
    {
        res = res + temp.getNum();
        temp = temp.getNext();
    }
    return res;
}

public int g() {
    int res = 0;
    IntNodeTwo temp = _head;
    while (temp != null)
    {
        res++;
        temp = temp.getNext();
    }
    return res;
}

public int what()
{
    int temp = g();
    int res = f();
    if (res%2 == 0)
        return temp;
    else
    {
        int i1 = 1;
        int i2 = 1;
        IntNodeTwo p1 = _head;
        IntNodeTwo p2 = _tail;
        while ((p1 != null) && (p1.getNum()%2 == 0))
        {
            i1++;
            p1 = p1.getNext();
        }
        while ((p2 != null) && (p2.getNum()%2 == 0))
        {
            i2++;
            p2 = p2.getPrev();
        }
        if (i1 < i2)
            return temp - i1;
        else
            return temp - i2;
    }
}

} //end of class IntListTwo

```

סעיף א (1 נקודות)

מה הערך אותו תחזיר השיטה f כשנפעיל אותה על הרשימה $\{3, 6, 2, 7\}$?
התשובה היא

סעיף ב (2 נקודות)

מה מבצעת השיטה f באופן כללי? הסבירו בקצרה מה השיטה עושה ולא **כיצד** היא מבצעת זאת.
שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה f , כשהיא מופעלת על רשימה כלשהי.

סעיף ג (1 נקודות)

מה הערך אותו תחזיר השיטה g כשנפעיל אותה על הרשימה $\{3, 6, 2, 7\}$?
התשובה היא

סעיף ד (2 נקודות)

מה מבצעת השיטה g באופן כללי? הסבירו בקצרה מה השיטה עושה ולא **כיצד** היא מבצעת זאת.
שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה g , כשהיא מופעלת על רשימה כלשהי.

סעיף ה (5 נקודות)

מה הערך אותו תחזיר השיטה **what** כשנפעיל אותה על הרשימה $\{3, 6, 2, 1, 4, 7, 8\}$?
התשובה היא

סעיף ו (8 נקודות)

מה מבצעת השיטה **what** באופן כללי? הסבירו בקצרה מה השיטה עושה ולא **כיצד** היא מבצעת זאת.
שימו לב, עליכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. כלומר, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה **what**, כשהיא מופעלת על רשימה כלשהי. התייחסו למקרי קצה.

ב ה צ ל ח ה