



מס' שאלון - 463
22 באפריל 2025

מס' מועד 75

סמסטר 2025א

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 19 עמודים

מבנה הבחינה:

- קראו בעיון את ההנחיות שלהלן:
- * עליכם לענות על חמש מתוך שש השאלות, בהתאם לסמסטר הלימוד שלכם.
 - * כל התכניות צריכות להיות מתועדות היטב.
 - יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.
 - יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתכנית.
 - תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85% מהניקוד.
 - * יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.
 - תכנית לא יעילה לא תקבל את מלוא הנקודות.
 - * אם ברצונכם להשתמש בתשובתכם בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיקו 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.
 - * בכתובת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה
 - אין להשתמש במשתנים גלובליים!
 - * אפשר לתעד בעברית. אין צורך בתיעוד API.
 - * על שאלות 3-5 יש לענות אך ורק בשאלון ולא במחברת הבחינה!

חומר עזר:

חוברות השקפים 1-6, 7-12. אסור לכתוב כלום בתוך חוברות השקפים.
מותר לסמן עמודים בצבע או בדגלונים. אסור להכניס מחשב/מחשבון/
מכשיר אלקטרוני מכל סוג שהוא. אסורים עזרים וחומרים מקוונים.
אין להכניס חומר נוסף אחר מכל סוג. חומר העזר מודפס בלבד.

בהצלחה !!!

חלק א – עליכם לענות על כל השאלות בחלק זה בקובץ הפתרונות

שאלה 1 (25 נקודות)

נתון מערך דו-ממדי mat המכיל את המספרים השלמים $0, 1$ או -1 בלבד. המספר -1 מסמן תא חסום. אנחנו מחפשים מסלול שמתחיל בתא הראשון במערך (שורה ראשונה ועמודה ראשונה) $mat[0][0]$. המסלול צריך לעבור אך ורק דרך תאים שאינם חסומים (כלומר, רק אם הערך שלהם הוא 0 או 1), המעבר מתא אחד לשני במסלול הוא לפי התנאים הבאים:

- אם הערך בתא (i, j) הוא 0 , אפשר לעבור לאחד מארבעת התאים **מימין** (שורה i ועמודה $j+1$), **משמאל** (שורה i ועמודה $j-1$), **למעלה** (שורה $i-1$ ועמודה j) או **למטה** (שורה $i+1$ ועמודה j);
- אם הערך בתא (i, j) הוא 1 , אפשר לעבור לאחד מארבעת התאים השכנים באלכסונים: (שורה $i+1$ ועמודה $j+1$), (שורה $i+1$ ועמודה $j-1$), (שורה $i-1$ ועמודה $j+1$) או (שורה $i-1$ ועמודה $j-1$);

המסלול יכול לבקר בכל תא במערך פעם אחת בלבד!

כתבנו כאן את כל ארבע האפשרויות בכל אחד מהמצבים, אבל לא בהכרח יש אכן את כל השכנים האלו. למשל, לתא בשורה 0 ובעמודה 0 , יש רק 3 שכנים, כך שאם יש בתא הזה 0 , אפשר לעבור ממנו לתאים מימין $(0,1)$ או למטה $(1,0)$, ואם יש בתא את הערך 1 , אפשר לעבור ממנו אך ורק לתא שבאלכסון $(1,1)$.

יתכן שהמסלול יגיע לתא האחרון במערך (שורה אחרונה ועמודה אחרונה), אך בהחלט יתכן שייתקע קודם לכן.

כתבו שיטה סטטית רקורסיבית, המקבלת כפרמטר מערך דו-ממדי המכיל את המספרים $0, 1$ ו- -1 בלבד, ומחזירה את אורך המסלול הארוך ביותר שקיים במערך, המתחיל בתחילת המערך, המקיים את תנאי המעבר המתוארים לעיל.

לדוגמא, אם נעביר את המערך שמצד ימין, כפרמטר לשיטה, הערך שהשיטה תחזיר הוא 7 . במערך שבצד שמאל אפשר לראות את המסלול באורך המקסימלי.

	0	1	2	3
0	1	1	0	1
1	1	0	0	-1
2	1	-1	1	-1

	0	1	2	3
0	1	1	0	1
1	1	0	0	-1
2	1	-1	1	-1

המסלולים האפשריים במערך הם אלו (אנו סופרים את התאים שמבקרים בהם):

1. $(0,0) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (1,2) \rightarrow (0,2) \rightarrow (0,3)$ - אורך 6
2. $(0,0) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (1,2) \rightarrow (2,2)$ - אורך 5
3. $(0,0) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (0,2) \rightarrow (0,3)$ - אורך 5
4. $(0,0) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (2,2)$ - אורך 4
5. **$(0,0) \rightarrow (1,1) \rightarrow (1,0) \rightarrow (0,1) \rightarrow (1,2) \rightarrow (0,2) \rightarrow (0,3)$** - אורך 7
6. $(0,0) \rightarrow (1,1) \rightarrow (1,0) \rightarrow (0,1) \rightarrow (1,2) \rightarrow (2,2)$ - אורך 6

אפשר לראות שהמסלול הארוך ביותר נתקע בשורה 0 עמודה 3, ולא הגיע לסוף המערך. למעשה אף מסלול לא הגיע לתא האחרון במערך (שורה אחרונה ועמודה אחרונה), שכן הערך בתא זה הוא 1- ולכן אינו יכול להשתתף באף מסלול.

חתימת השיטה היא:

```
public static int longestPath (int [][] mat)
```

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

- **מותר לשנות את המערך במהלך השיטה, אבל המערך צריך לחזור לקדמותו לאחר ביצוע השיטה.**
- **אסור ליצור מערך נוסף!**
- **מותר להשתמש בשיטות `Math.max` ו-`Math.min` מהמחלקה `Math` וכן בקבועים `Integer.MIN_VALUE`, `Integer.MAX_VALUE` מהמחלקה `Integer`.**
- **מותר להשתמש בהעמסת-יתר (`Overloading`)**
- **אין צורך לדאוג ליעילות השיטה! אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות!**
- **אל תשכחו לתעד את מה שכתבתם!**

שאלה 2 (25 נקודות)

נתון מערך חד-ממדי arr המלא במספרים שלמים.
נגדיר **תת-מערך** (sub-array) כאוסף של תאים **רצופים** בתוך המערך.

לדוגמא,

אם המערך הוא {1, 2, 3, 4}

אזי התת-מערכים שלו הם:

{1}, {1, 2}, {1, 2, 3}, {1, 2, 3, 4}

{2}, {2, 3}, {2, 3, 4}

{3}, {3, 4}

{4}

ובסה"כ 10 תת-מערכים (שאינם ריקים).

כתבו שיטה סטטית יעילה שמקבלת כפרמטרים מערך חד-ממדי arr המכיל מספרים שלמים (חיוביים שליליים ואפסים) ומספר שלם num.
השיטה צריכה להחזיר כמה תת-מערכים יש למערך arr שכל האיברים שלהם קטנים או שווים ל- num.

דוגמאות:

- אם המערך arr הוא {3, 20, 12, 5, 9, 8, 15, 2, 7} והמשתנה $num = 9$, השיטה תחזיר את הערך 10.

שכן אלו עשרת התת-מערכים שמכילים רק מספרים שקטנים או שווים ל-9.

{3}, {8, 9, 5}, {8, 9}, {5}, {9}, {8}, {2, 7}, {2}, {7}

(כדאי לשים לב לסדר בו כתבנו את התת-מערכים. עשוי לרמוז על דרך הפתרון)

- אם המערך arr נשאר כשהיה {3, 20, 12, 5, 9, 8, 15, 2, 7} אבל $num = 1$, השיטה תחזיר את הערך 0.

שכן אין במערך arr אף איבר שקטן או שווה ל-1, ולכן גם אין אף תת-מערך שכל איבריו קטנים או שווים ל-1.

חתימת השיטה היא:

```
public static int countSubArrays (int [] arr, int num)
```

שימו לב:

- המערך אינו ממוין.
- אפשר להניח שכל המספרים במערך שונים זה מזה.
- אינכם צריכים להדפיס את התת-מערכים, אלא רק להחזיר כמה יש.

השיטה שתכתבו צריכה להיות בסיבוכיות מקום קבועה (כלומר $O(1)$), ויעילה ככל הניתן מבחינת סיבוכיות הזמן. תשובה שאינה יעילה מספיק, כלומר, שתהיה בסיבוכיות זמן גדולה יותר מזו הנדרשת לפתרון הבעיה, תקבל מעט נקודות בלבד.

ציינו מהי סיבוכיות זמן הריצה ומהי סיבוכיות המקום של השיטה שכתבתם. הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!

חלק ב - את התשובות לשאלות 3, 4, 5 ו-6 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!

שאלה 3 (16 נקודות)

נניח שהמחלקה Node שלהלן מממשת צומת עץ בינרי שערכיו הם מספרים שלמים (int). שימו לב שבנוסף לשני המצביעים לבן השמאלי ולבן הימני, יש לכל אובייקט במחלקה Node גם שדה father שמצביע על האב של הצומת. בשורש העץ, המצביע לאב הוא null.

```
public class Node
{
    private int _data;
    private Node _leftSon, _rightSon, _father;

    public Node (int num) {
        _data = num;
        _leftSon = null;
        _rightSon = null;
    }

    public int getData()           { return _data; }
    public Node getLeftSon()       { return _leftSon; }
    public Node getRightSon()      { return _rightSon; }
    public Node getFather()        { return _father; }
}
```

המחלקה BinarySearchTree מאגדת בתוכה שיטות סטטיות לטיפול **בעץ חיפוש בינרי**. נתונות **ארבע** השיטות הבאות: maxInTree, g, what, secret ו-something המקבלות כפרמטר שורש (או צומת) של **עץ חיפוש בינרי** (מטיפוס Node):

```
public class BinarySearchTree {
    public static Node maxInTree(Node root)
    // השיטה מחזירה את הצומת המחזיק את הערך המקסימלי בעץ
    {
        if (root.getRightSon() == null)
            return root;
        return maxInTree (root.getRightSon());
    }
}
```

- המשך המחלקה בעמוד הבא -

```

public static Node g (Node node) {
    if (node.getFather() == null)
        return null;
    if (node.getFather().getRightSon() == node)
        return node.getFather();
    return g(node.getFather());
}

public static Node what (Node node) {
    if (node == null)
        return null;
    if (node.getLeftSon() != null)
        return maxInTree (node.getLeftSon());
    return g (node);
}

public static boolean secret(Node root) {
    if (root == null)
        return true;

    Node temp1 = what(root);
    if (temp1 == null)
        return true;

    Node temp2 = what(temp1);
    if (temp2 == null)
        return true;

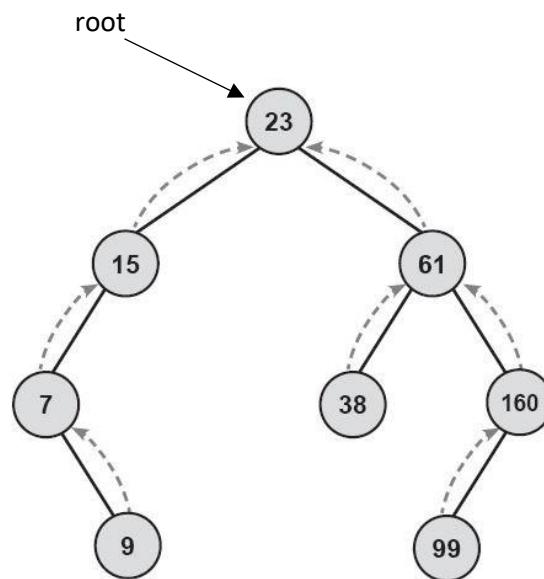
    if (root.getData() != temp1.getData() + temp2.getData())
    {
        System.out.println(root.getData() + " != " +
            temp1.getData() + " + " + temp2.getData());
        return false;
    }

    return secret(temp1);
}

public static boolean something(Node root) {
    return secret (maxInTree (root));
}
} // end of class BinarySearchTree

```

נתון עץ חיפוש בינרי ששורשו root המצויר להלן:



ענו על הסעיפים הבאים:

סעיף א (5 נקודות):

- אם נריץ את השיטה what כאשר הפרמטר node מצביע על הצומת שערכו 99, השיטה תחזיר מצביע על הצומת שערכו הוא _____
- אם נריץ את השיטה what כאשר הפרמטר node מצביע על הצומת שערכו 7, השיטה תחזיר מצביע על הצומת שערכו הוא _____
- אם בעץ לעיל, נוסיף לצומת שערכו 38 בן ימני שערכו 38 (וכמובן נסמן כ-father של הצומת החדש את הצומת המקורי שערכו 38), ואז נריץ את השיטה what כאשר הפרמטר node מצביע על הצומת החדש שערכו 38, השיטה תחזיר מצביע על הצומת שערכו הוא _____
- **טענה:** אם נריץ את השיטה what כאשר הפרמטר node מצביע על צומת כלשהו בעץ חיפוש בינרי, שערכו הוא num, וכן, נדפיס את צומתי העץ בסדר-תוכי (inOrder), השיטה תחזיר את הצומת בעץ שערכו הוא המספר הקודם לפני num בהדפסה זו. (2 נקודות)

האם הטענה נכונה? כן / לא (סמנו בעיגול בצורה ברורה).

אם התשובה היא "לא", הסבירו מה שגוי ומה צריך להיות כתוב כדי שהטענה תהיה נכונה.

סעיף ב (3 נקודות):

איזה ערך תחזיר השיטה **something** ומה היא תדפיס על הפלט, אם נקרא לה עם הפרמטר root שהוא שורשו של העץ לעיל?

התשובה היא:

סעיף ג (4 נקודות):

איזה שינוי מינימלי אפשר לעשות על העץ לעיל, כדי שהקריאה לשיטה (root) something תחזיר ערך אחר מזה שהוחזר בסעיף ב? שימו לב, השינוי צריך להיות בעץ ולא בשיטה (כגון, הוספת צומת, הורדת צומת, שינוי ערך של צומת וכד'). מינימלי במובן של מינימום פעולות על העץ. כך, אם מורידים צומת, ויש לצומת הזה בנים, במספר הפעולות נספרים גם הבנים של הצומת שהורדו.

התשובה היא:

סעיף ד (4 נקודות):

מה מבצעת השיטה **something** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ חיפוש בינרי root? 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, מה המשמעות של הערך שהשיטה מחזירה? התייחסו למקרי קצה.

התשובה היא:

שאלה 4 (16 נקודות)

נתונה המחלקה `IntNode` הבאה, המייצגת איבר ברשימה מקושרת חד-סטרית המכילה מספרים שלמים:

```
public class IntNode
{
    private int _value;
    private IntNode _next;

    public IntNode(int val, IntNode n) {
        _value = val;
        _next = n;
    }

    public int getValue()           { return _value; }
    public IntNode getNext()        { return _next; }
    public void setValue(int v)     { _value = v;   }
    public void setNext(IntNode node) { _next = node; }
}
```

נתונה רשימה מקושרת חד-סטרית, הממומשת בעזרת המחלקה `IntList` שמכילה את השיטה **what** הכתובה להלן.

הניחו שיש שיטות נוספות במחלקה, כגון שיטה להוספת איבר לרשימה, שיטה למחיקת איבר מהרשימה, וכן השיטה `toString` המחזירה מחרוזת תווים המייצגת את איברי הרשימה.

בשאלות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים. כאשר המספר **השמאלי** מסמן את **ראש** הרשימה.

```
public class IntList
{
    private IntNode _head;

    public IntList()           { _head = null; }
    public IntList(IntNode node) { _head = node; }
}
```

- - - המשך המחלקה בעמוד הבא - - -

```

public IntList what()
{
    IntNode head = _head;
    if (head == null) {
        return null;
    }

    IntNode p1 = head;
    IntNode p2 = head.getNext();

    while (p2 != null)
    {
        if (p1.getValue() > p2.getValue())
        {
            int temp = p1.getValue();
            p1.setValue(p2.getValue());
            p2.setValue(temp);
        }

        if (p2.getNext() != null &&
            p2.getNext().getValue() > p2.getValue())
        {
            int temp = p2.getNext().getValue();
            p2.getNext().setValue(p2.getValue());
            p2.setValue(temp);
        }

        p1 = p2.getNext();

        if (p2.getNext() != null)
            p2 = p2.getNext().getNext();
        else
            p2 = p2.getNext();
    }

    return new IntList(head);
}
//end of class IntList

```

סעיף א (3 נקודות)

נפעיל את השיטה what על הרשימה list הבאה:

{1, 2, 3, 4, 5, 6, 7, 8}, מה יודפס לאחר ביצוע הפקודה

```
System.out.println (list.what());
```

התשובה היא:

סעיף ב (3 נקודות)

נפעיל את השיטה what על הרשימה list הבאה:

{8, 7, 6, 5, 4, 3, 2, 1}, מה יודפס לאחר ביצוע הפקודה

```
System.out.println (list.what());
```

התשובה היא:

סעיף ג (3 נקודות)

האם יש רשימה list2 בת 7 איברים לפחות, כאשר כל האיברים שונים זה מזה, כך שאם נפעיל את השיטה what על הרשימה, הדפסת הרשימה לפני ואחרי הפעלת השיטה תהיה זהה?

אם יש כזו, כתבו דוגמא לרשימה כזו.

אם לא, נמקו מדוע.

התשובה היא:

סעיף ד (3 נקודות)

האם יש רשימה list2 בת 7 איברים לפחות, כאשר כל שני איברים סמוכים שונים זה מזה, כך שאם נפעיל את השיטה what על הרשימה, הדפסת הרשימה אחרי הפעלת השיטה תהיה הפוכה להדפסה לפני ההפעלה?

אם יש כזו, כתבו דוגמא לרשימה כזו.

אם לא, נמקו מדוע.

התשובה היא:

סעיף ה (4 נקודות)

מה מבצעת השיטה **what** באופן כללי, כאשר היא פועלת על רשימה list כלשהי? 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.
התייחסו למקרי קצה.

התשובה היא:

לפניך שתי שאלות. יש לענות על שאלה אחת מבין השתיים בהתאם לסמסטר הלימוד שלך.

- אם למדת את הקורס בסמסטר 2024 א או 2024 ב – חובה עליך לענות על שאלה 5
- אם למדת את הקורס בסמסטר 2025 א או לפני 2024 – חובה עליך לענות על שאלה 6

לתשומת לבך! באחריות הסטודנט/ית להקפיד על בחירת השאלה המתאימה. שאלה לא מתאימה לא תיבדק!

שאלה 5 (18 נקודות) – רק למי שלמד/ה את הקורס בסמסטר 2024 א/ב

במחלקה Q5 נתונות השיטות הסטטיות הבאות:

secret, maximum, minimum ו-secret. השיטה secret מקבלת 3 מערכים חד-ממדיים באותו אורך, המכילים מספרים שלמים. כל אחד מהמערכים ממוין בסדר לא-יורד:

```
public class Q5
{
    static int maximum(int a, int b, int c)
    {
        return Math.max(Math.max(a, b), c);
    }

    static int minimum(int a, int b, int c)
    {
        return Math.min(Math.min(a, b), c);
    }
}
```

- המשך המחלקה בעמוד הבא -

```

public static void secret(int arr1[], int arr2[],
                          int arr3[])
{
    int temp1=0, temp2=0, temp3=0;
    int i = 0, j = 0, k = 0;
    int res = Integer.MAX_VALUE;
    int n = arr1.length;

    while (i < n && j < n && k < n)
    {
        int curr = arr1[i] + arr2[j] + arr3[k];
        int max = maximum(arr1[i], arr2[j], arr3[k]);
        int min = minimum(arr1[i], arr2[j], arr3[k]);
        if (min == arr1[i])
            i++;
        else if (min == arr2[j])
            j++;
        else
            k++;

        if (res > (max - min))
        {
            res = max - min;
            temp2 = max;
            temp3 = curr - (max + min);
            temp1 = min;
        }
    }

    System.out.print(temp2 + ", " + temp3 + ", " + temp1);
}

} //end of class Q5

```

סעיף א (4 נקודות):

אם נתונים המערכים הבאים:

```

int [] arr1 = {9, 12, 15, 18};
int [] arr2 = {-8, 10, 13, 17};
int [] arr3 = {-5, 11, 11, 16};

```

מה תדפיס השיטה `secret` בעקבות הקריאה `secret(arr1, arr2, arr3)`, בהינתן לה כפרמטרים המערכים שלעיל?

התשובה היא:

סעיף ב (4 נקודות)

תנו דוגמא לשלושה מערכים באורך 4 לפחות, ממוינים בסדר לא-יורד, שאם נעביר אותם כפרמטרים לשיטה secret יודפס הפלט 1, 2, 3.

התשובה היא:

סעיף ג (4 נקודות)

תנו דוגמא לשלושה מערכים באורך 4 לפחות, ממוינים בסדר **עולה ממש**, שאם נעביר אותם כפרמטרים לשיטה secret יודפס הפלט -10, -10, -10.

התשובה היא:

סעיף ד (6 נקודות)

בניח שנתונים שלושה מערכים חד-ממדיים, כולם באורך שווה, שמלאים במספרים שלמים, כל אחד מהמערכים ממוין בסדר לא-יורד.

מה מבצעת השיטה secret אם נפעיל אותה עם המערכים האלו כפרמטרים? 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. **הסבירו מה המשמעות של כל אחד מהערכים המודפסים כתוצאה מהפעלת השיטה secret.** התייחסו למקרי קצה.

התשובה היא:

שאלה 6 (18 נקודות) - רק למי שלמד/ה את הקורס בסמסטר 2025 או לפני 2024

נתון פרויקט שהוגדרו בו המחלקות שלהלן. כל אחת בקובץ נפרד, כמובן.

```
public abstract class A {
    protected int _x;

    public A() {
        _x = 1;
    }

    public abstract int f(int x);
    public void f(A a) {
        _x = a._x;
    }
} //end of class A
```

```
-----
public class B extends A {
    public B() {
        super();
    }
    public B(int val) {
        _x = f(val);
    }

    public int f(int x) {
        return _x + x;
    }

    public void f(B b) {
        _x = _x * b._x;
    }
} //end of class B
```

```
-----
public class D extends B {
    public D(int val) {
        _x = val - _x;
    }

    public void f(A a) {
        _x = _x + a._x + 1;
    }

    public void f(B b) {
        _x = _x * b._x;
    }

    public void f(D d) {
        _x = d._x-1;
    }
} //end of class D
```

כמו כן נתונה המחלקה Driver הבאה באותו פרויקט:

```
public class Driver
{
    public static void main (String [] args)
    {
        A a;
        B b;

        // כאן יופיעו הפקודות שבסעיפים להלן
    }
}
```

עליכם לענות על חמשת הסעיפים הבאים.

שימו לב שהשאלות אינן תלויות אחת בשניה!

סעיף א (3 נקודות) –

נניח שנתונה הפקודה הבאה, כאשר m הוא משתנה מטיפוס `int`:

```
a = new D(m);
```

מה צריך להיות ערכו של הפרמטר m כדי שלאחר ביצוע הפקודה, הערך של התכונה `_x` של

האובייקט `a` יהיה 10?

התשובה היא:

סעיף ב (3 נקודות) –

נניח שנתונות שתי הפקודות הבאות:

```
b = new B(3);
```

```
int k = b.f(b.f(4));
```

מה יהיה הערך של המשתנה `k` לאחר ביצוע הפקודות?

התשובה היא:

סעיף ג (3 נקודות) –

נניח שנתונות שתי הפקודות הבאות:

```
a = new B(5);  
int k = a.f(a.f(2));
```

מה יהיה ערכו של המשתנה k לאחר ביצוע שתי הפקודות?

התשובה היא:

סעיף ד (4 נקודות) –

נניח שנתונות שלוש הפקודות הבאות:

```
a = new D(10);  
b = new B(20);  
a.f(b);
```

מה יהיה ערכה של התכונה x של האובייקט a לאחר ביצוע שלוש הפקודות?

התשובה היא:

סעיף ה (5 נקודות) –

נניח שנתונות שלוש הפקודות הבאות, כאשר m הוא משתנה כלשהו מסוג int:

```
D d = new D(m);  
b = new B();  
(A)d.f(b);
```

מה צריך להיות ערכו של הפרמטר m כדי שלאחר ביצוע שלוש הפקודות הערך של התכונה

x של האובייקט d יהיה 5?

התשובה היא:

ב ה צ ל ח ה !