

סוג הבחינה: א. בגרות לבתי ספר על-יסודיים
ב. בגרות לנבחנים חיצוניים
מועד הבחינה: קיץ תשע"ג, 2013
מספר השאלון: 603,899205

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.
פרק ראשון – בפרק זה ארבע שאלות, — (25x2) — 50 נקודות
ומהן יש לענות על שתיים.
פרק שני – בפרק זה שאלות בארבעה מסלולים שונים. — (25x2) — 50 נקודות
ענה על שאלות רק במסלול שלמדת,
לפי הוראות בקבוצת השאלות במסלול זה.
סה"כ — 100 נקודות
- ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.
- ד. הוראות מיוחדות:
(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון
כתוב בשפה אחת בלבד – C# או Java.
(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – C# או Java.
(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת,
אחד מארבעת המסלולים: מערכות מחשב ואסמבלר, מבוא לחקר ביצועים,
מודלים חישוביים, תכנות מונחה עצמים.
הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום
אות קטנה או להפך.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

/המשך מעבר לדף/

השאלות

בשאלון זה שני פרקים: פרק ראשון ופרק שני.
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.
פרק ראשון (50 נקודות)

שים לב: בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות `Node<T>`, `List<T>`, `BinTreeNode<T>`, בלי לממש אותן. אם אתה משתמש בפעולות נוספות עליך לממש אותן.

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. שים לב: לשאלה זו שני נוסחים: אחד ב-Java (עמודים 2-3), ואחד ב-C# (עמודים 4-5). עבוד על פי השפה שלמדת.

לפותרים ב-Java

לפניך שלוש פעולות `sod1`, `sod2`, `sod3`.

```
public static boolean sod1(List<Integer> list1, List<Integer> list2)
{
    Node<Integer> node1 = list1.getFirst();
    Node<Integer> node2 = list2.getFirst();
    for (int i = 1; i <= 4; i++)
    {
        if ((node1 == null) || (node2 == null))
            return false;
        if ((i == 1) || (i == 4))
            if (node1.getInfo() != node2.getInfo())
                return false;
        node1 = node1.getNext();
    }
    return true;
}
```

```
public static boolean sod2(List<Integer> list1, List<Integer> list2)
{
    Node<Integer> node1 = list1.getFirst();
    Node<Integer> node2 = list2.getFirst();
    while ((node1 != null) && (node2 != null))
    {
        if (node1.getInfo() != node2.getInfo())
            return false;
        node1 = node1.getNext();
        node2 = node2.getNext();
    }
    return true;
}
```

/המשך בעמוד 3/

```

public static boolean sod3(List<Integer> list1, List<Integer> list2)
{
    Node<Integer> node1 = list1.getFirst();
    while (node1 != null)
    {
        boolean found = false;
        Node<Integer> node2 = list2.getFirst();
        while ((node2 != null) && (!found))
        {
            if (node1.getInfo() == node2.getInfo())
                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 יחזור בעבור כל אחת מהפעולות. במעקב הראה את המעבר על הרשימות.

ב. מהי סיבוכיות זמן הריצה של כל אחת מהפעולות sod1, sod2, sod3? נמק את תשובתך.

לפותרים ב- C#

לפניך שלוש פעולות Sod1, Sod2, Sod3.

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

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

```

public static bool Sod3(List<int> list1, List<int> list2)
{
    Node<int> node1 = list1.GetFirst();
    while (node1 != null)
    {
        bool found = false;
        Node<int> node2 = list2.GetFirst();
        while ((node2 != null) && (!found))
        {
            if (node1.GetInfo() == node2.GetInfo())
                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 זמן הריצה של כל אחת מהפעולות Sod1, Sod2, Sod3? נמק את תשובתך.

2. **שים לב:** לשאלה זו שני נוסחים: אחד ב־ Java (עמוד 6), ואחד ב־ C# (עמוד 7). עבוד על פי השפה שלמדת.

לפותרים ב־ Java

עץ טרינארי הוא עץ שיש בו לכל היותר שלושה בנים לצומת, בן שמאלי, בן אמצעי ובן ימני. לפניך ממשק חלקי של המחלקה "צומת טרינארי" `TriTreeNode<T>`. המחלקה מגדירה צומת טרינארי שבו ערך מטיפוס `T` ושלוש הפניות לצמתים טרינאריים.

שם הפעולה	תיאור
<code>TriTreeNode(T x)</code>	הפעולה בונה צומת טרינארי שערכו יהיה <code>x</code> , וערך שלוש ההפניות יהיה <code>null</code> .
<code>TriTreeNode(TriTreeNode<T> left, TriTreeNode<T> middle, TriTreeNode<T> 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 ההפניות יכולים להיות <code>null</code> .
<code>T getInfo()</code>	הפעולה מחזירה את הערך של הצומת הטרינארי.
<code>void setInfo(T x)</code>	הפעולה משנה את הערך השמור בצומת הטרינארי ל־ <code>x</code> .
<code>TriTreeNode<T> getLeft()</code>	הפעולה מחזירה את הבן השמאלי של הצומת הטרינארי. אם אין בן שמאלי, הפעולה מחזירה <code>null</code> .
<code>TriTreeNode<T> getMiddle()</code>	הפעולה מחזירה את הבן האמצעי של הצומת הטרינארי. אם אין בן אמצעי, הפעולה מחזירה <code>null</code> .
<code>TriTreeNode<T> getRight()</code>	הפעולה מחזירה את הבן הימני של הצומת הטרינארי. אם אין בן ימני, הפעולה מחזירה <code>null</code> .

א. כתוב ב־ Java 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 boolean noThree(TriTreeNode<Integer> 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 תחזיר `true`.

ממש את הפעולה ב־ Java.

הערה: אתה יכול להשתמש בפעולות המחלקה `TriTreeNode` בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן. /המשך בעמוד 7/

C# לפותרים ב-

עץ טרינארי הוא עץ שיש בו לכל היותר שלושה בנים לצומת, בן שמאלי, בן אמצעי ובן ימני. לפניך ממשק חלקי של המחלקה "צומת טרינארי" `TriTreeNode<T>`. המחלקה מגדירה צומת טרינארי שבו ערך מטיפוס `T` ושלוש הפניות לצמתים טרינאריים.

שם הפעולה	תיאור
<code>TriTreeNode(T x)</code>	הפעולה בונה צומת טרינארי שערכו יהיה <code>x</code> , וערך שלוש ההפניות יהיה <code>null</code> .
<code>TriTreeNode(TriTreeNode<T> left, TriTreeNode<T> middle, TriTreeNode<T> 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 ההפניות יכולות להיות <code>null</code> .
<code>T GetInfo()</code>	הפעולה מחזירה את הערך של הצומת הטרנארי.
<code>void SetInfo(T x)</code>	הפעולה משנה את הערך השמור בצומת הטרנארי ל- <code>x</code> .
<code>TriTreeNode<T> GetLeft()</code>	הפעולה מחזירה את הבן השמאלי של הצומת הטרנארי. אם אין בן שמאלי, הפעולה מחזירה <code>null</code> .
<code>TriTreeNode<T> GetMiddle()</code>	הפעולה מחזירה את הבן האמצעי של הצומת הטרנארי. אם אין בן אמצעי, הפעולה מחזירה <code>null</code> .
<code>TriTreeNode<T> GetRight()</code>	הפעולה מחזירה את הבן הימני של הצומת הטרנארי. אם אין בן ימני, הפעולה מחזירה <code>null</code> .

א. כתוב ב- C# פעולה חיזונית `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(TriTreeNode<int> 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 תחזיר `true`. ממש את הפעולה ב- C#.

הערה: אתה יכול להשתמש בפעולות המחלקה `TriTreeNode` בלי לממש אותן.

אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

3. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

- א. לאבן משחק יש שני צדדים. על צד אחד רשום אחד מהמספרים החד-ספרתיים בין 0 ל-6 (כולל), ועל הצד האחר רשום אחד מהמספרים הדו-ספרתיים בין 10 ל-16 (כולל).
- (1) כתוב ב-Java או ב-C# את כותרת המחלקה **BiStone**, המגדירה אבן משחק, ואת התכונות שלה.
- (2) כתוב ב-Java או ב-C# במחלקה **BiStone**, פעולה בונה המקבלת שני מספרים, האחד חד-ספרתי בין 0 ל-6, והאחר דו-ספרתי בין 10 ל-16.
- (3) כתוב ב-Java או ב-C# 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 – הפעולה תחזיר false.

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

- (4) כתוב ב-Java או ב-C# את כותרת המחלקה **Stones** ואת התכונות שלה.
- (5) כתוב ב-Java או ב-C# במחלקה **Stones**, פעולה בונה שתיצור את כל 49 אבני ה-**BiStone**.

ב. (אין קשר לסעיף א.)

- (1) כתוב ב-Java או ב-C# 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**.
- (2) מהי סיבוכיות זמן הריצה של הפעולה שכתבת? נמק את תשובתך.

שיום לב: לשאלה זו שני נוסחים: אחד ב־ Java (עמודים 9-10), ואחד ב־ C# (עמודים 11-12). עבוד על פי השפה שלמדת.

.4

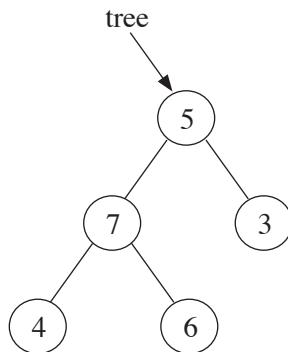
לפותרים ב־ Java

בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך הפעולה `amir`.

```
public static void amir(BinTreeNode<Integer> t, int x)
{
    if (t != null)
    {
        amir(t.getLeft() , x + 1);
        if ((t.getLeft() == null) && (t.getRight() == null))
            t.setInfo(x) ;
        amir(t.getRight() , x + 1);
    }
}
```

לפניך העץ `tree`.



i רשום את העץ `tree` שיתקבל לאחר הפעלת הזימון: `amir(tree , 4)`.

ii מה מבצעת הפעולה `amir` בעבור עץ כלשהו ר־ $x = 0$?

(שיום לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

נתון עץ חיפוש בינארי t , המכיל לפחות שלושה צמתים שערכיהם: x, y, z .
לפניך שתי טענות:

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

(2) אם x, y, z 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.

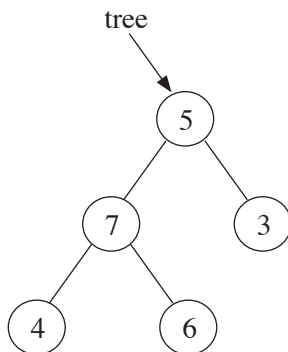
לפתרים ב- C#

בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך הפעולה Amir .

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

לפניך העץ tree .



- i רשום את העץ tree שיתקבל לאחר הפעלת הזימון: Amir(tree , 4) .
- ii מה מבצעת הפעולה Amir בעבור עץ כלשהו ו- $x = 0$?

(שים לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

נתון עץ חיפוש בינארי t , המכיל לפחות שלושה צמתים שערכיהם: x, y, z .
לפניך שתי טענות:

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

(2) אם x, 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.

פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלר, עמ' 13-17.

מבוא לחקר ביצועים, עמ' 18-24.

מודלים חישוביים, עמ' 25-29.

תכנות מונחה עצמים ב־ Java, עמ' 30-36; תכנות מונחה עצמים ב־ C#, עמ' 37-43.

ענה רק על שאלות במסלול שלמדת.

מערכות מחשב ואסמבלר

אם למדת מסלול זה, ענה על שתיים מהשאלות 5-8 (לכל שאלה — 25 נקודות).

5. במקטע הנתונים הוגדרו הנתונים בצורה זו:

```
NIBBLE    DB    1001B
M          DW    0F000H
NUMBER    DW    0E539H
```

לפניך קטע תכנית באסמבלר.

```
MOV CX , 4
XOR DX , DX
MOV DH , NIBBLE
SHL DH , CL
MOV CX , 13
MOV AX , NUMBER
AGAIN: MOV BX , M
      AND BX , AX
      XOR BX , DX
      JZ    OK
      SHL AX , 1
      LOOP AGAIN
      JMP  SOF
OK:   MOV CH , 1
SOF:  NOP
```

א. עקוב בעזרת טבלת מעקב אחר הביצוע של קטע התכנית.

בטבלת המעקב פֶּרְט בכל שלב את התוכן של AX , BX , CX , DX .

ב. מה מבצע קטע התכנית?

6. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. במקטע הנתונים הוגדר המשתנה K בצורה זו:

```
K    DB 5EH
```

לפניך קטע תכנית באסמבלר.

```
MOV CX , 2
MOV AH , K
MOV AL , AH
NEXT:  SHL  AX , CL
       DEC  CL
       SHR  AH , CL
       ADD  AH , AL
       SUB  AH , CL
       LOOP NEXT
```

עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית.

בטבלת המעקב פֶּרְט בכל שלב את התוכן של: AL, AH, CL.

ב. (אין קשר לסעיף א.)

לפניך 4 קטעי תכנית באסמבלר, iv-i. לאחר ביצוע כל אחד מהקטעים, על האוגר AX להכיל את אותו ערך שהכיל לפני ביצוע הקטע. בעבור כל אחד מהקטעים iv-i שלפניך, קבע אם הוא מבצע את הנדרש או אינו מבצע את הנדרש. אם הקטע אינו מבצע את הנדרש, הבא דוגמה המראה זאת.

i MOV CL , 4
 ROL AX , CL
 PUSH AX
 POP BX
 ROR BX , CL
 MOV AX , BX

ii MOV CX , 4
 XOR BX , BX
 MOV BX , AX
 SHR AX , CL
 SHL BX , CL
 MOV AX , BX

iii MOV CX , 4
 XOR BX , BX
 ADD BX , AX
 SHL BX , CL
 SHR BX , CL
 MOV AX , BX

iv MOV CX , 4
 ROL AX , CL
 PUSH AX
 POP DX
 ROR DX , CL
 MOV AX , DX

7. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך קטע תכנית הכתוב ב- Java וב- C#.

```
while ((a<b) || (a > 0))
    {a = a-1;}
```

המשתנים a ו- b הם משתנים מטיפוס שלם.

כתוב קטע תכנית באסמבלר שיבצע את קטע התכנית שלפניך.

הנח שהמשתנה a מאוחסן באוגר AX, והמשתנה b מאוחסן באוגר BX.

ב. (אין קשר לסעיף א.)

(1) רשום הוראה אחת שתאפס את שני הביטים הראשונים ואת שני הביטים האחרונים

של האוגר AL, מבלי לשנות את הערכים של הביטים האחרים באוגר.

(2) רשום הוראה אחת שתדליק (תהפוך ל- 1) את שני הביטים הראשונים ואת

שני הביטים האחרונים של האוגר AL, מבלי לשנות את הערכים של הביטים האחרים באוגר.

(3) רשום הוראה אחת שתגרום שארבעת הביטים הנמוכים של האוגר AL יהיו 1,

מבלי לשנות את הערכים של ארבעת הביטים הגבוהים של האוגר AL.

(4) במקטע הנתונים הוגדר המערך A בצורה זו:

```
A DW 20 DUP (?)
```

— רשום הוראה אחת שתכניס לאוגר AX את התוכן של האיבר החמישי

של המערך A.

— רשום הוראה אחת שתכניס לאוגר AX את הכתובת של האיבר החמישי של

המערך A.

8. במקטע הנתונים הוגדרו הנתונים האלה:

ARR DW 100 DUP(?)

MIN DW ?

NUM DW 0

כתוב באסמבלר קטע תכנית שיציב במשתנה MIN את הערך הקטן ביותר המאוחסן במערך ARR. קטע התכנית יציב במשתנה NUM את מספר הפעמים שהערך, שנמצא במשתנה MIN, הופיע במערך.

על קטע התכנית לעבור פעם אחת בלבד על כל האיברים של המערך ARR. עליך להתייחס לכל המספרים כמספרים לא מסומנים.

/המשך בעמוד 18/

מבוא לחקר ביצועים

אם למדת מסלול זה, ענה על שתיים מהשאלות 9-12 (לכל שאלה – 25 נקודות).

9. נתונה בעיית תכנון לינארי:

$$\max \{z = kx_1 + 6x_2\}$$

בכפוף לאילוצים האלה:

$$2x_1 + 3x_2 \leq 6$$

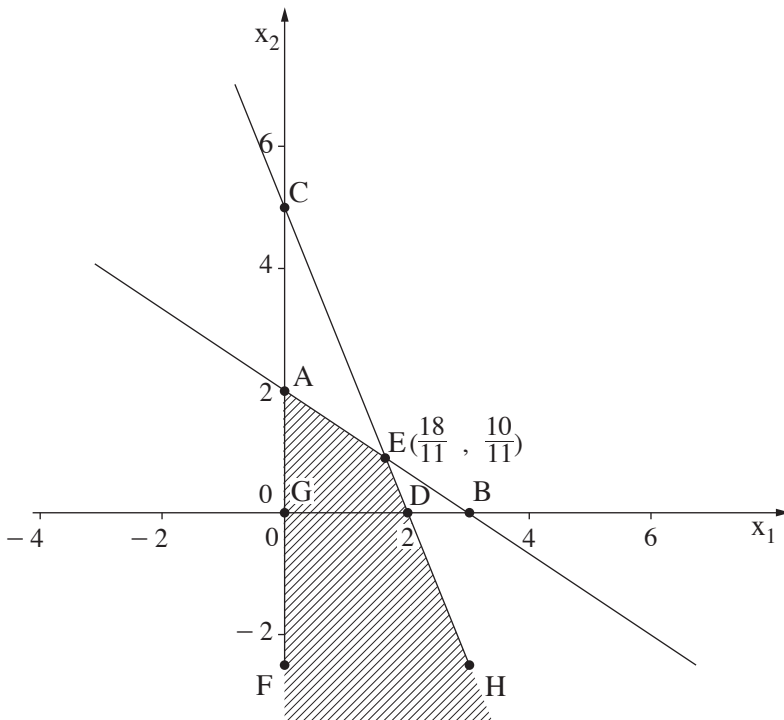
$$5x_1 + 2x_2 \leq 10$$

$$x_1 \geq 0$$

x_2 אינו מוגבל בסימן.

k הוא פרמטר.

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



א. לפניך חמישה תת-סעיפים (1)-(5) שאינם תלויים זה בזה. בחלק מהם נתון ערך מסוים של הפרמטר k .

בעבור כל אחד מהתת-סעיפים (1)-(5):

- אם הפתרון האופטימלי הוא יחיד, מצא את הפתרון האופטימלי היחיד, ואת הערך של פונקציית המטרה בפתרון זה.
- אם הפתרון האופטימלי אינו חסום, נמק את קביעתך.
- אם אין פתרון אפשרי, נמק את קביעתך.

(1) $k = 0$.

(2) $k = 6$.

(3) משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל:
 $\min \{z = 6x_2\}$.

(4) משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל:
 $\min \{z = 6x_1 + 6x_2\}$.

(5) משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל:
 $\min \{z = 6x_1 + 6x_2\}$, ומוסיפים את האילוץ $x_2 \geq 0$.

ב. בעבור אילו ערכים של k יהיו לבעיית התכנון הלינארי הנתונה בתחילת השאלה אין-סוף פתרונות אופטימליים?

קבע את הערך של פונקציית המטרה בעבור כל אחד מערכי ה- k האלה. נמק את קביעתך.

10. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.
 א. יהי $G = (V, E)$ גרף לא מכוון המיוצג על ידי מטריצת הסמיכויות שלפניך:

	a	b	c	d	e
a	0	1	1	0	0
b	1	0	1	0	0
c	1	1	0	1	0
d	0	0	1	0	1
e	0	0	0	1	0

- (1) סרטט את הגרף G המיוצג על ידי המטריצה.
- (2) כמה רכיבי קשירות (Connected Components) יש בגרף הנתון, ומה הם?
- (3) הסר קשת אחת מהגרף G שסרטטת, כך שמספר רכיבי הקשירות לא ישתנה. ציין את הקשת שהסרת.
- (4) קשת בגרף תיקרא "קשת מפרידה" אם הסרתה מהגרף תגדיל את מספר רכיבי הקשירות ב-1.
 ציין "קשת מפרידה" אחת בגרף G .

סרטט את הגרף שמתקבל לאחר הסרת הקשת שציינת.

ב. (אין קשר לסעיף א.)

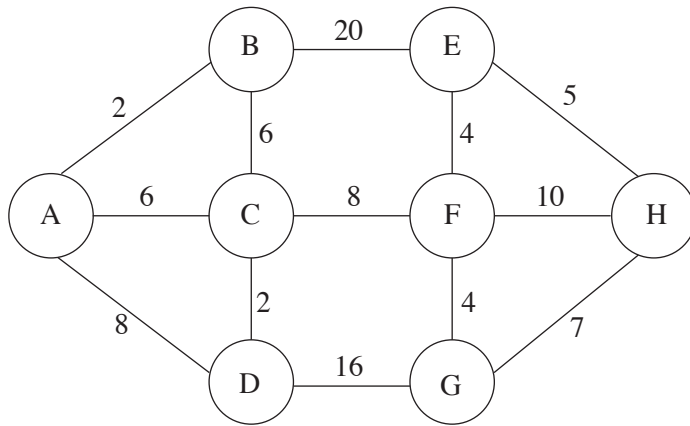
יהי $G = (V, E)$ גרף לא מכוון המיוצג על ידי רשימת הסמיכויות שלפניך:

a	→ b → c → g → e →
b	→ a → e → c →
c	→ a → b → f →
d	→ e →
e	→ a → b → d →
f	→ c → g →
g	→ a → f →

- (1) הפעל אלגוריתם סריקה לעומק (DFS) על הגרף הנתון, החל מקדקוד a. סרטט במחברתך רק את העץ הפורש (DFS) / היער הפורש (DFS) שמתקבל. התבסס על הייצוג הנתון על ידי רשימת הסמיכויות.
- (2) הפעל אלגוריתם סריקה לרוחב (BFS) על הגרף הנתון, החל מקדקוד a. סרטט במחברתך רק את העץ הפורש (BFS) / היער הפורש (BFS) שמתקבל. התבסס על הייצוג הנתון על ידי רשימת הסמיכויות.

11. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

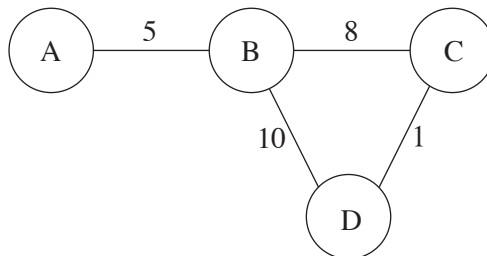
א. לפניך רשת:



מצא את המסלולים הקצרים ברשת הנתונה לפי האלגוריתם של דיקסטר, מקדקוד A לכל קדקוד אחר ברשת, וצייר את התיאור הסכמתי של עץ המסלולים הקצרים. בעבור כל קדקוד ברשת, רשום את אורך המסלול הקצר ביותר מקדקוד A עד אליו.

ב. (אין קשר לסעיף א.)

לפניך רשת:



(1) מצא את המסלולים הקצרים ברשת הנתונה לפי האלגוריתם של דיקסטר, מקדקוד A לכל קדקוד אחר ברשת, וצייר את התיאור הסכמתי של עץ המסלולים הקצרים.

בעבור כל קדקוד ברשת, רשום את אורך המסלול הקצר ביותר מקדקוד A עד אליו.

(2) מוסיפים קשת לא מכוונת (A, D) לרשת הנתונה.

מה צריך להיות משקלה של הקשת, כדי שהמסלולים הקצרים מקדקוד A לכל אחד מהקדקודים האחרים לא ישתנו?

12. בשאלה זו תמישה סעיפים, א-ה, שאין קשר ביניהם. ענה על כל הסעיפים.

א. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית התובלה: $x_{11} = 5$.

מקורות	יעדים				היצע
	1	2	3	4	
1	10 5	10	20	11	15
2	12	7	19	20	25
3	0	14	16	18	5
ביקוש	5	15	15	10	

i העתק את הטבלה למחברתך והשלם את הפתרון הבסיסי האפשרי לפי שיטת הפינה הצפונית מערבית.

ii משנים את המחיר בתא (3, 1) מ-0 ל-15. האם ישתנה הפתרון הבסיסי האפשרי שמצאת בתת-סעיף i? נמק.

ב. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה ונתון ערכו של u_1 .

מקורות	יעדים				היצע	u_i
	1	2	3	4		
1	10 5	10 5	24	11 5	15	0
2	12	7 10	19 15	20	25	
3	0	24	26	18 5	5	
ביקוש	5	15	15	10		
v_j						

i העתק את הטבלה למחברתך והשלם בה את הערכים של $u_2, u_3, v_1, v_2, v_3, v_4$.

ii הסבר מדוע הפתרון הנתון אינו פתרון אופטימלי.

ג. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכיהם של $u_1, u_2, u_3, v_1, v_2, v_3, v_4$ שמתאימים לפתרון זה.

מקורות	יעדים				היצע	u_i
	1	2	3	4		
1	10	10	20	11	15	-7
		5		10		
2	12	7	19	20	25	-10
		10	15			
3	0	14	16	18	5	0
	5					
ביקוש	5	15	15	10		
v_j	0	17	29	18		

העתק את הטבלה למחברתך והשלם את הפתרון תוך התחשבות בערכים של $u_1, u_2, u_3, v_1, v_2, v_3, v_4$ כך שיתקבל פתרון בסיסי אפשרי.

ד. נתונה טבלה אשר מייצגת בעיית תובלה מסוימת.

מקורות	יעדים				היצע
	A	B	C	D	
1	10	10	20	11	15
2	12	7	19	20	25
3	0	14	16	18	5
ביקוש	5	15	15	10	

משנים את הביקוש של היעד C מ-15 ל-20, בלי לשנות את שאר הביקושים וההיצעים שבטבלה. סרטט במחברתך טבלה חדשה אשר תכלול את כל הנתונים שבטבלה הנתונה, ותאפשר למצוא פתרון בסיסי אפשרי לפי שיטת הפינה הצפונית מערבית. בטבלה יש לכלול את המחירים מכל מקור לכל יעד.

(שים לב: סעיף ה של השאלה בעמוד הבא.)

ה. בטבלה שלפניך נתון פתרון לא אופטימלי שהתקבל לאחר k איטרציות בעבור בעיית תובלה מסוימת, ונתונים ערכיהם של $u_1, u_2, u_3, v_1, v_2, v_3, v_4$.

מקורות	יעדים				היצע	u_i
	1	2	3	4		
1	10 6	10 9	20	31	15	0
2	12	7 5	19 15	20 3	23	-3
3	0	14	20	18 7	7	-5
ביקוש	6	14	15	10		
v_j	10	10	22	23		

i מהו המשתנה היוצא מהבסיס?

ii סרטט במחברתך טבלה חדשה ובה הפתרון שיתקבל לאחר עוד איטרציה, כלומר לאחר האיטרציה ה- $k+1$.

/המשך בעמוד 25/

מודלים חישוביים

אם למדת מסלול זה, ענה על שתיים מהשאלות 13-16 (לכל שאלה – 25 נקודות).

13. בשאלה זו שני סעיפים א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך השפה L מעל הא"ב $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$:

מילה שייכת לשפה L אם מכפלת כל הספרות המרכיבות את המילה מתחלקת ב-10 ללא שארית.

דוגמאות למילים ב' L:

325, כי $3 \cdot 2 \cdot 5 = 30$ מתחלק ב-10 ללא שארית.

30675, כי $3 \cdot 0 \cdot 6 \cdot 7 \cdot 5 = 0$ מתחלק ב-10 ללא שארית.

543, כי $5 \cdot 4 \cdot 3 = 60$ מתחלק ב-10 ללא שארית.

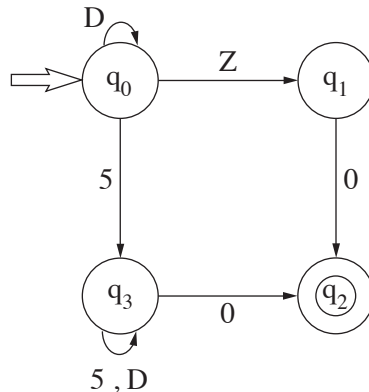
דוגמאות למילים שאינן ב' L:

123, כי $1 \cdot 2 \cdot 3 = 6$ אינו מתחלק ב-10 ללא שארית.

51, כי $5 \cdot 1 = 5$ אינו מתחלק ב-10 ללא שארית.

נסמן: $D = \{1, 3, 7, 9\}$ $Z = \{2, 4, 6, 8\}$.

לפניך סרטוט חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L. בסרטוט חסרים מעברים וסימני קלט.



הסרטוט מכיל את כל המצבים של האוטומט, ואת כל המצבים המקבלים.

העתק למחברתך את הסרטוט, והשלם אותו כך שהאוטומט יקבל את השפה L.

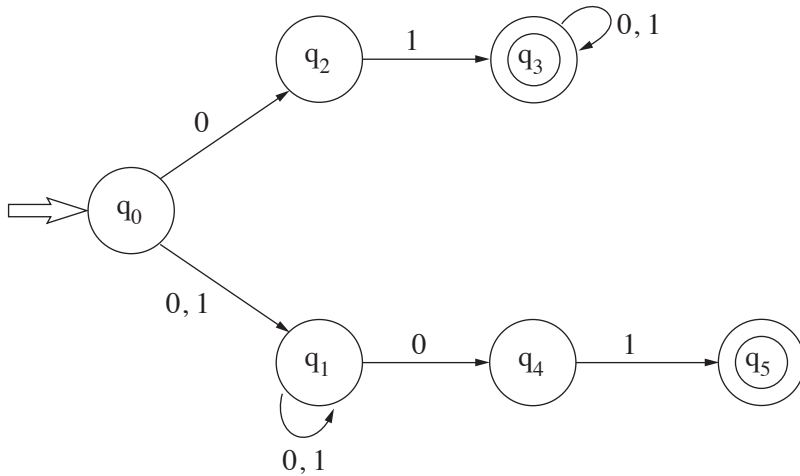
עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים.

שים לב: אין להוסיף מצבים לאוטומט ואין להוריד ממנו מצבים.

(שים לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך אוטומט סופי לא דטרמיניסטי המקבל את השפה L מעל הא"ב $\{0, 1\}$.



(1) העתק למחברתך כל אחת מחמש המילים $v-i$ שלפניך.

לכל אחת מהמילים קבע אם היא מתקבלת על ידי האוטומט או אינה מתקבלת.

אם המילה מתקבלת על ידי האוטומט, רשום במחברתך את המסלול המקבל

בעבור מילה זו.

- 000 i
- 01000 ii
- 1100 iii
- 00101 iv
- 1011 v

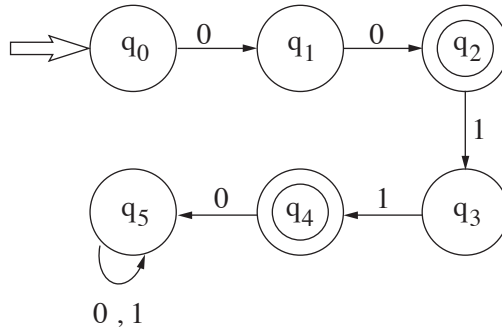
(2) מהי השפה L המוגדרת על ידי האוטומט הנתון?

14. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך השפה L מעל הא"ב $\{0, 1\}$:

$$L = \{0^n 1^k \mid n > 0, k \geq 0, n + k \text{ זוגי}\}$$

לפניך סרטוט חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L .
בסרטוט חסרים מעברים וסימני קלט.



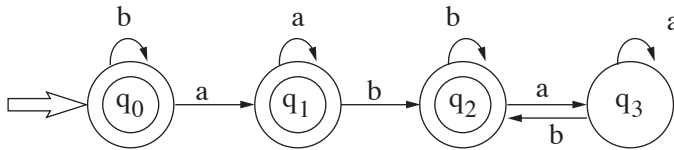
הסרטוט מכיל את כל המצבים של האוטומט, ואת כל המצבים המקבלים.

העתק למחרתך את הסרטוט, והשלם אותו כך שהאוטומט יקבל את השפה L .
עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים.
שים לב: אין להוסיף מצבים לאוטומט, ואין להוריד ממנו מצבים.

(שים לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך אוטומט סופי דטרמיניסטי המקבל את השפה L מעל הא"ב $\{a, b\}$.



לכל אחת מהטענות i - iii שלפניך קבע אם היא נכונה או אינה נכונה.

אם הטענה אינה נכונה כתוב מילה המפריכה אותה.

i. L היא אוסף כל המילים מעל הא"ב $\{a, b\}$ שמכילות את התת־מילה ab

ואינן מסתיימות ב־ b .

ii. L היא אוסף כל המילים מעל הא"ב $\{a, b\}$ אשר מסתיימות ב־ b

ואינן מכילות את התת־מילה ab .

iii. L היא אוסף כל המילים מעל הא"ב $\{a, b\}$ אשר מסתיימות ב־ b

או אינן מכילות את התת־מילה ab .

15. L היא שפה מעל הא"ב $\{a, b, c\}$.

$$L = \{a^n b^n c^k \mid n \geq 1, k \geq 3\}$$

בנה אוטומט מחסנית בעבור השפה L .

/המשך בעמוד 29/

16. בשאלה זו שלושה סעיפים, א-ג, שאין קשר ביניהם. ענה על שלושתם.

א. לפניך השפות L_1 , L_2 מעל הא"ב $\{a, b\}$:

$$L_1 = \{v \cdot v^R \mid v \in \{a, b\}^*\}$$

$$L_2 = \{a^m b \mid m \geq 1\}$$

$\{a, b\}^*$ הוא אוסף כל המילים מעל הא"ב $\{a, b\}$ (כולל המילה הריקה),

v^R היא ההיפוך של המילה v .

נתונות שתי מילים ב- L_2 : $w_1 = a^i b$ ו- $w_2 = a^j b$ כך ש: $j \neq i$.

מצא מילה w כך שיתקיים:

$$w_1 \cdot w \in L_1$$

$$w_2 \cdot w \notin L_1$$

ב. לפניך השפה L_1 מעל הא"ב $\{a, b, c\}$, והשפה L_2 מעל הא"ב $\{a\}$:

$$L_1 = \{a^n b^k c^m \mid k = \max(n, m), n \geq 1, m \geq 1\}$$

$$L_2 = \{a^s \mid s \geq 1\}$$

נתונות שתי מילים ב- L_2 : $w_1 = a^i$, $w_2 = a^j$, כך ש: $j > i$.

מצא מילה w כך שיתקיים:

$$w_1 \cdot w \in L_1$$

$$w_2 \cdot w \notin L_1$$

ג. לפניך השפה L_1 מעל הא"ב $\{a, b\}$, והשפה L_2 מעל הא"ב $\{a, b, c\}$:

$$L_1 = \{a^n b^k \mid n \geq 0, k \geq 0\}$$

$$L_2 = \{a^m c b^r \mid m + r \text{ ב-} 2 \text{ היא } 1\} \cup \{a^n b^n \mid n \geq 0\}$$

(1) בנה אוטומט סופי דטרמיניסטי המקבל את השפה L_1 .

(2) מהי השפה $L_1 \cap L_2$? האם היא רגולרית? נמק.

(3) היעזר בתוצאה שקיבלת בתת-סעיף (2), והסבר מדוע השפה L_2 אינה רגולרית.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב־ Java, ענה על שתיים מהשאלות 17-20. (לכל שאלה — 25 נקודות)

17. א. לפניך חלק מהממשק של המחלקה Node, ובו ארבע פעולות.

Node(T x)	הפעולה בונה חוליה. הערך של החוליה הוא x, ואין לה חוליה עוקבת.
Node(T x , Node<T> next)	הפעולה בונה חוליה. הערך של החוליה הוא x, והחוליה העוקבת לה היא next. ערכו של next יכול להיות null.
Node<T> getNext()	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה null.
String toString()	הפעולה מחזירה מחרוזת המתארת את החוליה.

- (1) מהו המנגנון המאפשר שתי פעולות ששמן זהה באותה מחלקה?
- (2) האם אפשר להוסיף לממשק המחלקה Node פעולה בונה שלישית? אם כן — כתוב כותרת מתאימה לפעולה, אם לא — הסבר מדוע.

ב. נתונות המחלקות Node1<T> , Node2<T> היורשות את המחלקה Node<T> .

```
public class Node1<T> extends Node<T>
{
    public Node1(T x)           { super(x); }
    public Node1(T x, Node<T> n) { super(x, n); }
    public String toString()     {return "$" + super.toString() + "$";}
}
public class Node2<T> extends Node<T>
{
    public Node2(T x)           { super(x); }
    public Node2(T x, Node<T> n) { super(x, n); }
    public String toString()     {return "#" + super.toString() + "#";}
}
```

נתונה הפעולה main במחלקה Test .

בפעולה main חסרים חלקי הוראות בשורות המסומנות (i)-(v) .

העתק שורות אלה למחברתך והשלם אותן, כך שפלט הפעולה main יהיה (משמאל לימין)

1\$2\$#3#\$4\$5

```
public class Test
{
    public static void main(String[] args)
    {
        Node<Integer> nd = _____ (i) _____ ;
        nd = _____ (ii) _____ ;
        nd = _____ (iii) _____ ;
        nd = _____ (iv) _____ ;
        nd = _____ (v) _____ ;

        while (nd != null)
        {
            System.out.print(nd);
            nd = nd.getNext();
        }
    }
}
```

18. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך הגדרות:

```
interface IOne {...}
interface ITwo {...}
interface IThree extends IOne {...}
class C1 implements IOne {...}
class C2 implements ITwo {...}
class C3 implements IThree {...}
class C4 extends C3 implements ITwo {...}
```

לפניך שמונה קטעי תכנית, i-viii. קבע אם כל אחד מהם תקין או אינו תקין.
אם אינו תקין – נמק את קביעתך, וקבע אם יש שגיאת הידור (קומפילציה) או שגיאת ריצה.

```
i      ITwo a = new ITwo();
ii     ITwo b = new C2();
iii    C3 c = new C4();
iv     C2 d = new C4();
v      C4 e = new C3();
vi     C4 f = (C4)(new C3());
vii    IOne g1 = new C1();
        C4 g2 = new C4();
        g1 = g2;
viii   IOne h1 = new C4();
        ITwo h2 = new C2();
        h2 = h1;
```


ב. (אין קשר לסעיף א.)

נתונות שתי מחלקות A, B, ונתונה הפעולה main במחלקה Test.

```
public class A
{
    public A(int k) { System.out.print(k + " "); }
}
public class B extends A
{
    public B(int k)
    {
        A a = new A(10);
        System.out.print(k + " ");
    }
}
public class Test
{
    public static void main(String[] args) { A a = new B(6); }
}
```

המחלקה B אינה עוברת קומפילציה.

i הסבר מהי השגיאה.

ii תקן את השגיאה שהסברת בתת-סעיף i כך שפלט הפעולה main יהיה (משמאל לימין) 6 10, מבלי להוסיף פקודות הדפסה או לשנות פקודות הדפסה קיימות, ומבלי לשנות את המחלקה Test.

iii תקן את השגיאה שהסברת בתת-סעיף i כך שפלט הפעולה main יהיה 6 10 6, מבלי להוסיף פקודות הדפסה או לשנות פקודות הדפסה קיימות, ומבלי לשנות את המחלקה Test.

העתק למחברתך את המחלקה / המחלקות שתיקנת.

19. בשאלה זו שני סעיפים, א-ב, שאינם קשורים זה לזה. ענה על שניהם.

א. לפניך פרויקט ובו המחלקות AAA, BBB ו- Run.

```
public class AAA
{
    protected int x;
    public AAA(int k)    { x = k;    }
    public String toString() {return "H"; }
}
public class BBB extends AAA
{
    public BBB(int k)    { super(k);    }
    public String toString()    {return "T";    }
    public AAA convertToAAA() {return new AAA(x);}
}
public class Run
{
    public static void main (String[] args)
    {
        AAA s1 = new AAA(10);
        AAA s2 = new BBB(20);
        BBB s3 = new BBB(30);
        System.out.print(s1 + " ");
        System.out.print(s2 + " ");
        System.out.print(s3 + " ");
        System.out.print(((AAA)s3 + " ");
        System.out.print(s3.convertToAAA());
    }
}
```

כתוב מעקב אחר הפעולה main במחלקה Run, וכתוב את הפלט.
 במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם — את ערכי התכונות שלו.

ב. (אין קשר לסעיף א.)

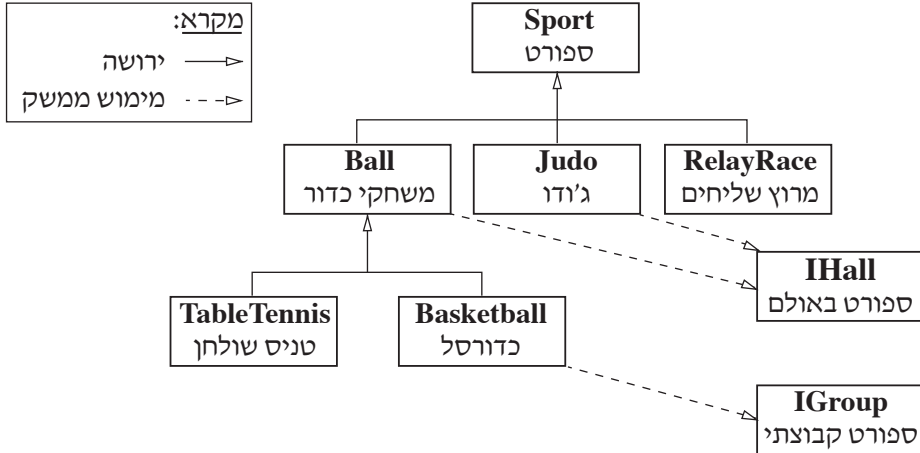
לפניך פרויקט ובו המחלקות A , B ו־ Test .

```
public class A
{
    public int x;
    public A()      { this(1);      }
    public A(int k) { this.x = k*10; }
    public int m()  { return x;     }
}
public class B extends A
{
    public int x;
    public B()      { this(2);      }
    public B(int k)
    {
        super(k + 1);
        this.x = super.x + 1;
    }
    public int m()  { return x;     }
}
public class Test
{
    public static void main(String[] args)
    {
        A a = new A();
        System.out.println(a.x);
        System.out.println(a.m());
        B b = new B();
        System.out.println(b.x);
        System.out.println(b.m());
        a = new B();
        System.out.println(a.x);
        System.out.println(a.m());
        b = (B)a;
        System.out.println(b.x);
        System.out.println(b.m());
    }
}
```

כתוב מעקב אחר הפעולה main במחלקה Test, וכתוב את הפלט.

במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם — את ערכי התכונות שלו.

20. לפניך תרשים של הייררכיית מחלקות בעבור המחלקות **Sport** – ספורט, **Ball** – משחקי כדור, **Judo** – ג'ודו, **RelayRace** – מרוץ שליחים, **Basketball** – כדורסל, **TableTennis** – טניס שולחן, והממשקים **IHall** – ספורט באולם ו- **IGroup** – ספורט קבוצתי.



א. (1) לפניך ארבע הגדרות, i-iv. קבע אם כל אחת מהן נכונה או אינה נכונה.
אם אינה נכונה, נמק את קביעתך.

- i Sport s1 = new Ball();
- ii Sport s2 = new Basketball();
- iii Judo b2 = new Sport();
- iv Basketball b3 = new Sport();

(2) לפניך שני היגדים, i-ii. קבע אם כל אחד מהם נכון או אינו נכון. נמק את קביעתך.

- i אפשר להגדיר במחלקה **Ball** משתנים מסוג `private` שאינם מוגדרים במחלקה **Sport** או במחלקה **Judo**.
- ii אם במחלקה **Sport** יש הגדרה של משתנה בשם `title`, בהכרח המחלקות **Ball** ו- **Judo** יורשות אותו, אך אינן יכולות לגשת אליו ישירות.

ב. כתוב כותרת ב- Java לכל מחלקה ולכל ממשק שבתרשים.

ג. כתוב ב- Java פעולה שחתימתה `public static int count(Sport[] sp)`, שתחזיר את מספר העצמים במערך `sp` המממשים את שני הממשקים.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- `C#`, ענה על שתיים מהשאלות 21-24.
(לכל שאלה – 25 נקודות)

21. א. לפניך חלק מהממשק של המחלקה `Node`, ובו ארבע פעולות.

<code>Node(T x)</code>	הפעולה בונה חוליה. הערך של החוליה הוא <code>x</code> , ואין לה חוליה עוקבת.
<code>Node(T x, Node<T> next)</code>	הפעולה בונה חוליה. הערך של החוליה הוא <code>x</code> , והחוליה העוקבת לה היא <code>next</code> . ערכו של <code>next</code> יכול להיות <code>null</code> .
<code>Node<T> GetNext()</code>	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה <code>null</code> .
<code>string ToString()</code>	הפעולה מחזירה מחרוזת המתארת את החוליה.

- (1) מהו המנגנון המאפשר שתי פעולות ששמן זהה באותה מחלקה?
- (2) האם אפשר להוסיף לממשק המחלקה `Node` פעולה בונה שלישית?
אם כן – כתוב כותרת מתאימה לפעולה, אם לא – הסבר מדוע.

(שים לב: סעיף ב בעמוד הבא.)

ג. נתונות המחלקות Node1<T>, Node2<T> היורשות את המחלקה Node<T>.

```
public class Node1<T> : Node<T>
{
    public Node1(T x) : base(x) { }
    public Node1(T x, Node<T> n):base(x,n) { }
    public override string ToString() { return "$" + base.ToString() + "$"; }
}

public class Node2<T> : Node<T>
{
    public Node2(T x) : base(x) { }
    public Node2(T x, Node<T> n):base(x,n) { }
    public override string ToString() { return "#" + base.ToString() + "#"; }
}
```

נתונה הפעולה Main במחלקה Test.

בפעולה Main חסרים חלקי הוראות בשורות המסומנות (i)-(v).

העתק שורות אלה למחברתך והשלם אותן, כך שפלט הפעולה Main יהיה (משמאל לימין)

1\$2\$#3#\$4\$5

```
public class Test
{
    public static void Main ()
    {
        Node<int> nd = _____ (i) _____;
        nd = _____ (ii) _____;
        nd = _____ (iii) _____;
        nd = _____ (iv) _____;
        nd = _____ (v) _____;
        while (nd != null)
        {
            Console.Write(nd);
            nd = nd.GetNext();
        }
    }
}
```

22. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך הגדרות:

```
interface IOne {...}
interface ITwo {...}
interface IThree : IOne {...}
class C1 : IOne {...}
class C2 : ITwo {...}
class C3 : IThree {...}
class C4 : C3 , ITwo {...}
```

לפניך שמונה קטעי תכנית, i-viii. קבע אם כל אחד מהם תקין או אינו תקין.
אם אינו תקין – נמק את קביעתך, וקבע אם יש שגיאת הידור (קומפילציה) או שגיאת ריצה.

```
i   ITwo a = new ITwo();
ii  ITwo b = new C2();
iii C3 c = new C4();
iv  C2 d = new C4();
v   C4 e = new C3();
vi  C4 f = (C4)(new C3());
vii IOne g1 = new C1();
     C4 g2 = new C4();
     g1 = g2;
viii IOne h1 = new C4();
      ITwo h2 = new C2();
      h2 = h1;
```

(שים לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

נתונות שתי מחלקות A, B, ונתונה הפעולה Main במחלקה Test.

```
public class A
{
    public A(int k) { Console.Write(k + " "); }
}

public class B : A
{
    public B(int k)
    {
        A a = new A(10);
        Console.Write(k + " ");
    }
}

public class Test
{
    public static void Main() { A a = new B(6); }
}
```

המחלקה B אינה עוברת קומפילציה.

i הסבר מהי השגיאה.

ii תקן את השגיאה שהסברת בתת-סעיף i כך שפלט הפעולה Main יהיה (משמאל לימין) 6 10 , מבלי להוסיף פקודות הדפסה או לשנות פקודות הדפסה קיימות, ומבלי לשנות את המחלקה Test.

iii תקן את השגיאה שהסברת בתת-סעיף i כך שפלט הפעולה Main יהיה 6 10 6 , מבלי להוסיף פקודות הדפסה או לשנות פקודות הדפסה קיימות, ומבלי לשנות את המחלקה Test.

העתק למחברתך את המחלקה / המחלקות שתיקנת.

23. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך פרויקט ובו המחלקות AAA , BBB ו- Run .

```
public class AAA
{
    protected int x;
    public AAA(int k)          { x = k;          }
    public override string ToString() { return "H"; }
}

public class BBB: AAA
{
    public BBB(int k): base(k)    {              }
    public override string ToString() {return "T";    }
    public AAA ConvertToAAA() {return new AAA(x);}
}

public class Run
{
    public static void Main()
    {
        AAA s1 = new AAA(10);
        AAA s2 = new BBB(20);
        BBB s3 = new BBB(30);
        Console.Write(s1 + " ");
        Console.Write(s2 + " ");
        Console.Write(s3 + " ");
        Console.Write(((AAA)s3 + " ");
        Console.Write(s3.ConvertToAAA());
    }
}
```

כתוב מעקב אחר הפעולה Main במחלקה Run , וכתוב את הפלט.
 במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם — את ערכי התכונות שלו.

(שים לב: סעיף ב בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך פרויקט ובו המחלקות A , B ו- Test .

```
public class A
{
    public int x;
    public A() : this(1)    { }
    public A(int k)        { this.x = k*10; }
    public virtual int M() { return x; }
}

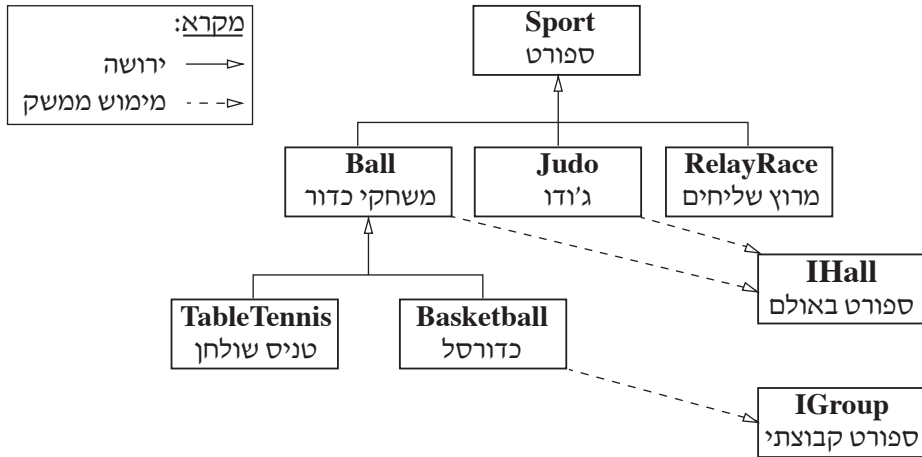
public class B : A
{
    public int x;
    public B() : this(2)    { }
    public B(int k) : base(k + 1) { this.x = base.x + 1; }
    public override int M() { return x; }
}

public class Test
{
    public static void Main()
    {
        A a = new A();
        Console.WriteLine(a.x);
        Console.WriteLine(a.M());
        B b = new B();
        Console.WriteLine(b.x);
        Console.WriteLine(b.M());
        a = new B();
        Console.WriteLine(a.x);
        Console.WriteLine(a.M());
        b = (B)a;
        Console.WriteLine(b.x);
        Console.WriteLine(b.M());
    }
}
```

כתוב מעקב אחר הפעולה Main במחלקה Test , וכתוב את הפלט.

במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם — את ערכי התכונות שלו.

24. לפניך תרשים של הייררכיית מחלקות בעבור המחלקות **Sport** – ספורט, **Ball** – משחקי כדור, **Judo** – ג'ודו, **RelayRace** – מרוץ שליחים, **BasketBall** – כדורסל, **TableTennis** – טניס שולחן, והממשקים **IHall** – ספורט באולם ו- **IGroup** – ספורט קבוצתי.



א. (1) לפניך ארבע הגדרות, i-iv. קבע אם כל אחת מהן נכונה או אינה נכונה.

אם אינה נכונה, נמק את קביעתך.

- i Sport s1 = new Ball();
- ii Sport s2 = new Basketball();
- iii Judo b2 = new Sport();
- iv Basketball b3 = new Sport();

(2) לפניך שני היגדים, i-ii. קבע אם כל אחד מהן נכון או אינו נכון. נמק את קביעתך.

- i אפשר להגדיר במחלקה **Ball** משתנים מסוג `private` שאינם מוגדרים במחלקה **Sport** או במחלקה **Judo**.
- ii אם במחלקה **Sport** יש הגדרה של משתנה בשם `title`, בהכרח המחלקות **Ball** ו- **Judo** יורשות אותו, אך אינן יכולות לגשת אליו ישירות.

ג. כתוב כותרת ב- C# לכל מחלקה ולכל ממשק שבתרשים.

ג. כתוב ב- C# פעולה שחתימתה `public static int Count(Sport[] sp)`, שתחזיר את מספר העצמים במערך `sp` המממשים את שני הממשקים.

בהצלחה!