

סוג הבחינה: א. בגרות לבתי ספר על-יסודיים
ב. בגרות לנבחנים חיצוניים
מועד הבחינה: קיץ תשע"ה, 2015
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, — (25×2) — 50 נקודות
ומהן יש לענות על שניים.
פרק שני — בפרק זה שאלות בארבעה מסלולים שונים. — (25×2) — 50 נקודות
ענה על שאלות רק במסלול שלמדת,
לפי ההוראות 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 — Java או C#.
 - (2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב — Java או C#.
 - (3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת.
המסלול הוא אחד מארבעת המסלולים האלה:
מערכות מחשב ואסמבלי, 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 רשימה, תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

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

1. **עץ ביטוי בוליאני** הוא עץ בינארי לא ריק מטיפוס מחרוזות, המייצג ביטוי בוליאני. בכל עלה שלו נמצאת אחת מן המחרוזות: "T" או "F". המחרוזת "T" מייצגת true, והמחרוזת "F" מייצגת false.

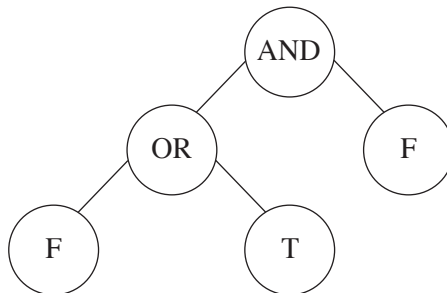
בכל צומת שאינו עלה נמצאת אחת מן המחרוזות: "AND" או "OR".

המחרוזת "AND" מייצגת את הפעולה הבוליאנית "וגם", והמחרוזת "OR" מייצגת את הפעולה הבוליאנית "או".

לכל צומת שאינו עלה יש שני בנים.

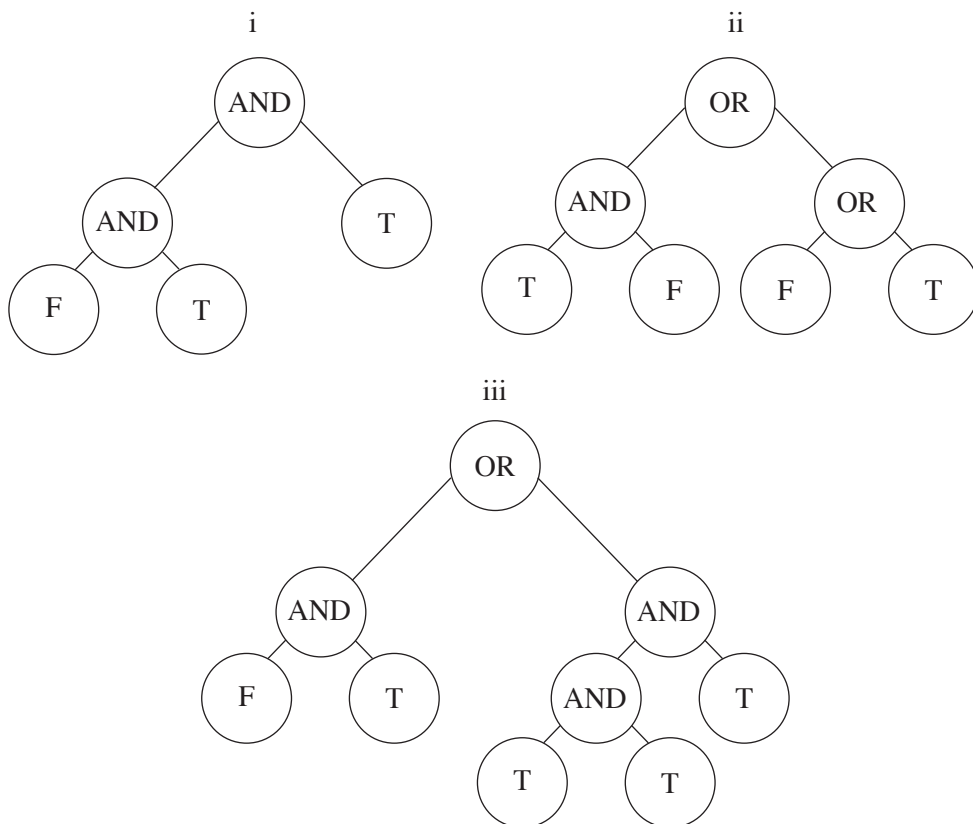
כדי לחשב את הביטוי הבוליאני שהעץ מייצג, מפעילים את הפעולה הבוליאנית שבצומת שאינו עלה על הערכים המתקבלים מהתת-עצים השמאלי והימני של צומת זה.

לדוגמה: הביטוי הבוליאני שמייצג העץ שלפניך הוא $((F \text{ OR } T) \text{ AND } (F))$ וערכו הוא false.



א. לפניך שלושה עצים i-iii שכל אחד מהם הוא עץ ביטוי בוליאני.

לכל אחד מן העצים כתוב את הביטוי הבוליאני שהוא מייצג, ואת הערך המתקבל ממנו.



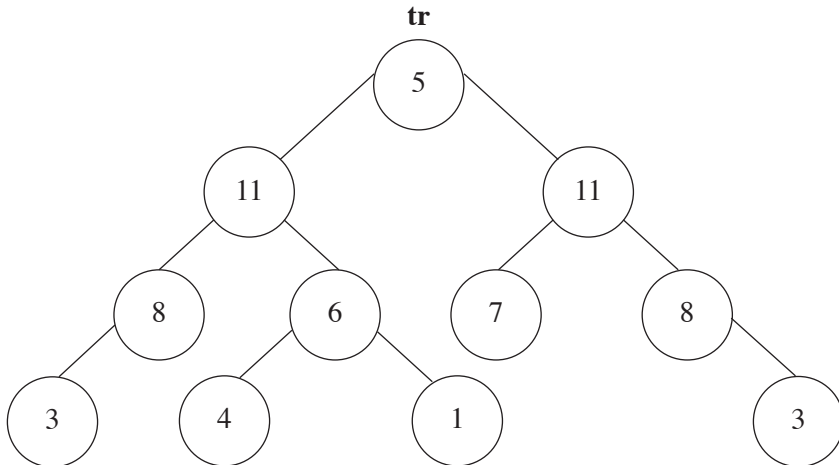
ב. כתוב ב-Java או ב-C# פעולה חיצונית שתקבל עץ ביטוי בוליאני, ותחזיר את

הערך הבוליאני (true או false) של הביטוי שהעץ מייצג.

/המשך בעמוד 4/

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

א. נתון עץ בינארי tr.



לפניך הפעולה wrap ב-Java המשתמשת בפעולה branch, והפעולה Wrap ב-C# המשתמשת בפעולה Branch.

שים לב: הפעולה כתובה בשני נוסחים. ענה על התת-סעיפים (1)-(2) שלפניך על פי הנוסח שלמדת.

הנוסח לפותרים על פי התכנית הנוכחית (ב-Java וב-C#) בעמוד 5,

הנוסח לפותרים על פי התכנית החדשה (ב-Java וב-C#) בעמוד 6.

(1) עקוב אחר ביצוע הפעולה wrap ב-Java או Wrap ב-C#, בעבור העץ tr

הנתון, ורשום את הערך שיוחזר.

הראה את המעקב הרקורסיבי.

(2) הבא דוגמה לעץ בינארי שבעבורו הפעולה wrap ב-Java או Wrap ב-C#

תחזיר true, ודוגמה לעץ בינארי שבעבורו הפעולה תחזיר false.

על כל אחד מהעצים להכיל 5 צמתים בדיוק.

/המשך בעמוד 5/

Java

```

public static boolean wrap(BinTreeNode<Integer> tr)
{
    return branch(tr.getLeft(), tr.getRight());
}
public static boolean branch(BinTreeNode<Integer> t1, BinTreeNode<Integer> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.getInfo() == t2.getInfo()) && branch(t1.getLeft(), t2.getRight()));
}

```

C#

```

public static bool Wrap(BinTreeNode<int> tr)
{
    return Branch(tr.GetLeft(), tr.GetRight());
}
public static bool Branch(BinTreeNode<int> t1, BinTreeNode<int> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.GetInfo() == t2.GetInfo()) && Branch(t1.GetLeft(), t2.GetRight()));
}

```

(שים לב: סעיף ב של השאלה בעמוד 7.)

Java

```
public static boolean wrap(BinNode< Integer > tr)
{
    return branch(tr.getLeft(), tr.getRight());
}
public static boolean branch(BinNode<Integer> t1, BinNode<Integer> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.getValue() == t2.getValue()) && branch(t1.getLeft(), t2.getRight()));
}
```

C#

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

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך קטע תכנית הכתוב ב־ Java וב־ C#.

נתון ש־ a הוא מערך מטיפוס שלם בגודל n .

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

(1) עקוב אחר קטע התכנית בעבור המערך a בגודל 4 שלפניך.

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

במעקב יש להראות את i ואת המערך בסיום כל איטרציה.

(2) מהי סיבוכיות זמן הריצה של קטע התכנית בעבור מערך בגודל n , ממוין בסדר עולה?

נמק את תשובתך.

(3) עקוב אחר קטע התכנית בעבור המערך a בגודל 4 שלפניך.

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

במעקב יש להראות את i ואת המערך בסיום כל איטרציה.

(4) מהי סיבוכיות זמן הריצה של קטע התכנית בעבור מערך בגודל n , ממוין בסדר יורד?

נמק את תשובתך.

/המשך בעמוד 8/

3.

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

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

האוטובוס נוסע מתחנה לתחנה, לפי סדר התחנות.

אורך המסלול הוא סכום המרחקים בין תחנה לתחנה.

הנח שנתונה המחלקה תחנה — **Station** שתכונותיה שני מספרים שלמים x ו- y המציינים את מיקומה של התחנה במישור.

במחלקה **Station** הוגדרו שתי הפעולות:

פעולה בונה `Station(int x, int y)`

פעולה שכותרתה:

ב-`Java`: `double distance(Station other)`

ב-`C#`: `double Distance(Station other)`

הפעולה מקבלת תחנה `other` ומחזירה את המרחק בין התחנה הנוכחית ובין התחנה `other`.

לפניך ממשק המחלקה מסלול-אוטובוס — **BusRoute** הכתוב ב-`Java` וב-`C#`:

Java

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

C#

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

(שים לב: סעיפי השאלה בעמוד הבא.)

- א. כתוב ב-Java או ב-C# את כותרת המחלקה **BusRoute** ואת התכונה/התכונות שלה. כתוב תיעוד לכל תכונה.
- ב. ממש ב-Java או ב-C# את הפעולה הבונה של המחלקה **BusRoute**.
- ג. ממש ב-Java או ב-C# את הפעולה המוסיפה תחנה למסלול-אוטובוס.
- ד. ממש ב-Java או ב-C# את הפעולה המחזירה את אורך המסלול-אוטובוס.
- ה. כתוב ב-Java או ב-C# בפעולה הראשית במחלקה Program קטע תכנית שייצור את המסלול-אוטובוס שבדוגמה שבתחילת השאלה, וידפיס את אורך המסלול-אוטובוס. הערה: אתה יכול להשתמש בפעולות המחלקה **Station** בלי לממש אותן.

/המשך בעמוד 10/

4.

בחברה "עבודה יעילה" יש משימות: בהולות, דחופות ורגילות. משימות בהולות מתבצעות ראשונות, אחריהן מתבצעות משימות דחופות ולבסוף משימות רגילות.

בחברה יש מערכת ממוחשבת **סדרן משימות**.

ההכנסה של המשימות **לסדרן המשימות** נעשית על פי הכללים האלה:

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

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

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

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

נתונה המחלקה **משימה** – Task, שיש לה שתי תכונות:

content – מחרוזת שהיא תיאור המשימה,

code – מספר שלם המייצג את המשימה: 1 מייצג משימה בהולה; 2 מייצג משימה דחופה; 3 מייצג משימה רגילה.

הנח שלכל תכונה יש פעולות `get` ו-`set` ב-Java או `Get` ו-`Set` ב-C#.

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

א. כתוב ב-Java או ב-C# את כותרת המחלקה **סדרן משימות** ואת התכונות שלה.

כתוב תיעוד לכל תכונה.

ב. כתוב ב-Java או ב-C# פעולה בונה ללא פרמטרים של המחלקה **סדרן משימות**.

ג. כתוב ב-Java או ב-C# במחלקה **סדרן משימות**, פעולה שתקבל **משימה** ותכניס אותה ל**סדרן המשימות**, בהתאם לכללים המתוארים בתחילת השאלה.

ד. כתוב ב-Java או ב-C# במחלקה **סדרן משימות**, פעולה שתוציא את **המשימה** הבאה לביצוע, ותחזיר אותה.

אם אין **משימה** ב**סדרן המשימות**, הפעולה תחזיר null.

פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלי, עמודים 11-14.

מבוא לחקר ביצועים, עמודים 15-25.

מודלים חישוביים, עמודים 26-30.

תכנות מונחה עצמים ב-Java, עמ' 31-40; תכנות מונחה עצמים ב-C#, עמודים 41-50.

ענה רק על שאלות במסלול שלמדת.

מערכות מחשב ואסמבלי

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

5. לפניך קטע תכנית באסמבלי.

```

MOV     BX , 3000H
MOV     CX , 5
AGAIN:  MOV     AL , CL
        CMP     AL , 3
        JNZ     NEXT      ;(*)
        MOV     [BX] , CL
        INC     BX
NEXT:    LOOPZ   AGAIN
SHOOV:  MOV     AL , CL
        CMP     AL , 3
        JNZ     NEXT1
        MOV     [BX] , CL
        INC     BX
NEXT1:  LOOP    SHOOV      ;(**)
        NOP
    
```

א. עקוב בעזרת טבלת מעקב אחר הביצוע של קטע התכנית. בטבלת המעקב יש לכלול

עמודה לכל אחד מהאוגרים: CL, BX, AL.

ענה על הסעיפים ב-ד שלפניך.

א. קשר בין הסעיפים, וכל אחד מהם מתייחס לקטע התכנית הכתוב בתחילת השאלה.

ב. קבע כמה בתים יירשמו בזיכרון, החל בכתובת 3000H, לאחר ביצוע קטע התכנית.

ג. מחקו את ההוראה המסומנת ב-(*). קבע מה יירשם בזיכרון, החל בכתובת 3000H,

עם סיום קטע התכנית בלי ההוראה שנמחקה.

ד. התייחס לקטע התכנית המופיע בתחילת השאלה, כולל ההוראה המסומנת ב-(*).

במקום השורה המסומנת ב-(**) נכתבה השורה NEXT1: LOOPNE SHOOV.

קבע מה יירשם בזיכרון, החל בכתובת 3000H, עם סיום קטע התכנית לאחר השינוי.

/המשך בעמוד 12/

6.

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

א. בטבלה שלפניך כתובות הוראות שהן קטע תכנית באסמבלי.

העתק את הטבלה למחברתך, והשלם בכל אחת מן העמודות את מצב הדגלים לאחר הביצוע של כל אחת מן ההוראות.

הנח שלפני הביצוע של קטע התכנית כל שלושת הדגלים SF, ZF, CF מאופסים, והערך של BL הוא -1.

הוראות	CF	ZF	SF
MOV AL, 3H			
CMP AL, 3H			
CMP AL, 2H			
CMP AL, 5H			
XOR AL, AL			
DEC AL			
MUL BL			

ב. (אין קשר לסעיף א.)

(1) כתוב באסמבלי שגרה (פרוצדורה) שתקבל כפרמטר, באמצעות המחסנית, מספר בין 0 ל-255.

השגרה תאחסן באוגר AX את מספר ה-1 שיש בייצוג הבינארי של המספר.

(2) נתון מערך ARR בן 100 איברים שכל אחד מהם בגודל בית. המערך מכיל מספרים.

כתוב באסמבלי קטע תכנית שיאחסן באוגר CX את מספר האיברים במערך שיש בייצוג הבינארי שלהם מספר אי-זוגי של 1-ים.
עליך להשתמש בשגרה שכתבת בתת-סעיף ב (1).

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

א. לכל אחד מן ההיגדים (1)-(10) שלפניך, קבע אם הוא נכון או אינו נכון. אם ההיגד אינו נכון, הסבר מדוע.

(1) לפניך קטע תכנית באסמבלי.

```
PUSH    AL
POP      AH
```

קטע התכנית מכניס את הערך שב-AL ל- AH.

(2) בשיטת המשלים ל-2, באמצעות 8 ביטים אפשר לייצג מספרים

בין -128 ל-+127.

(3) ההוראה שלפניך מעתיקה את הערך של הכתובת בזיכרון השמורה ב-AX לתוך BX.

```
MOV      BX, [AX]
```

(4) ההוראה CALL משנה את SI.

(5) ההוראה RET מגדילה את ערכו של SP.

(6) ההוראה RET מקטינה תמיד את ערכו של IP.

(7) לפניך קטע תכנית באסמבלי שבו השגרה (פרוצדורה) DONOTHING אינה מבצעת

דבר, ואינה משנה את ערכי האוגרים.

```
MOV      CX, NUMEROFTIMES
LP:      CALL    DONOTHING
        LOOP    LP
```

אם NUMEROFTIMES שווה ל-0, הפרוצדורה DONOTHING תיקרא

פעם אחת.

(8) ההוראה שלפניך מאפסת את BX.

```
XOR      BX, BX
```

(9) קטע התכנית שלפניך מאפס את DL.

```
MOV      CL, 8
SHR      DL, CL
```

(10) במקטע הנתונים הוגדרו המשתנים NUM1, NUM2 :

```
NUM1     DB 100
NUM2     DB -156
```

הייצוג שלהם בזיכרון הוא זהה.

(שים לב: סעיף ב של השאלה בעמוד הבא.)

ב. (אין קשר לסעיף א.)

לפניך קטע תכנית הכתוב באסמבלי.

```
MOV    AL, -1
MOV    DL, 00001111B    ;(*)
MOV    CL, 4
AND    AL, DL
SHR    AL, CL
```

(1) עקוב בעזרת טבלת מעקב אחר הביצוע של קטע התכנית, וכתוב מה יהיה הערך של AL

לאחר הביצוע. בטבלת המעקב יש לכלול עמודה לכל אחד מן האוגרים: AL, CL, DL.

(2) בשורה המסומנת ב- (*) במקום המספר 00001111B נכתב המספר 00001111.

האם קטע התכנית יתורגם בלי שגיאות לשפת מכונה? נמק את תשובתך.

8. א. כתוב באסמבלי שגרה (פרוצדורה) בשם FINDSECOND שתקבל, באמצעות המחסנית,

שני פרמטרים: מצביע למערך של בתים, ומספר האיברים במערך.

השגרה FINDSECOND תאחסן באוגר AX את הערך השני בגודלו במערך.

לדוגמה בעבור המערך:

03H	08H	0CH	01H	09H
-----	-----	-----	-----	-----

הערך 09H יאוחסן באוגר AX.

הנחות:

— המספרים במערך חיוביים.

— המספרים במערך שונים זה מזה.

— במערך יש לפחות שני מספרים.

ב. במקטע הנתונים הוגדר המערך ARR בן 100 איברים שכל אחד מהם בגודל בית. המערך

מכיל מספרים.

כתוב באסמבלי קטע תכנית שידפיס את המספר השני בגודלו במערך, בייצוגו על פי

בסיס 10.

הנח שהמספרים במערך חיוביים ושונים זה מזה.

עליך להשתמש בשגרה FINDSECOND שכתבת בסעיף א.

מבוא לחקר ביצועים

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

9. בשאלה זו שישה סעיפים, א-ו, שאין קשר ביניהם. ענה על כל הסעיפים.

א. נתונים האילוצים:

$$6x_1 + 3x_2 \leq 18$$

$$-4x_1 + 2x_2 \leq 4$$

$$x_1 + x_2 \geq 3$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

סרטט במחברתך את תחום הפתרונות האפשריים הנקבע על סמך האילוצים הנתונים.

סמן את תחום הפתרונות האפשריים של האילוצים,

ורשום בגרף שמתקבל את הערכים של x_1 ו- x_2 בעבור כל אחד מן הקדקודים של

תחום הפתרונות האפשריים.

כל אחד מן הסעיפים ב-ו מתייחס לבעיית התכנון הלינארי שלפניך.

בעיית התכנון הלינארי:

$$\max \{z = 20x_1 + 15x_2\}$$

בכפוף לאילוצים האלה:

$$2x_1 + x_2 \leq 6$$

$$-2x_1 + x_2 \leq 2$$

$$x_1 + x_2 \leq b$$

$$x_1 \geq 0$$

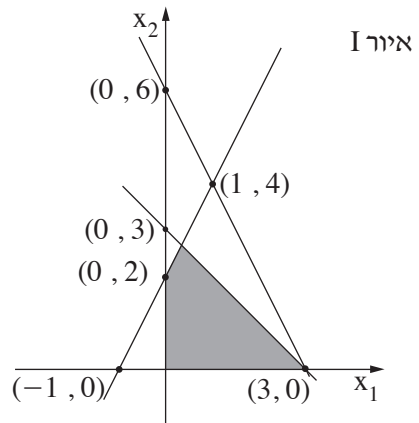
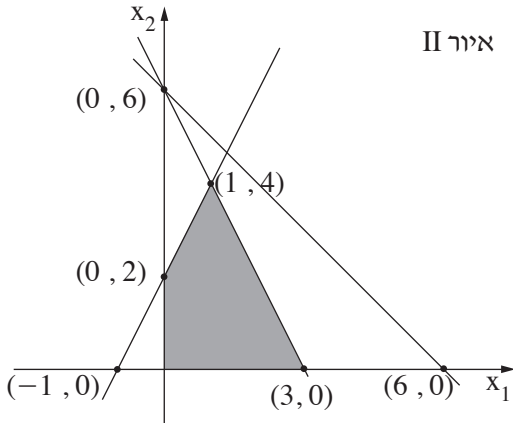
$$x_2 \geq 0$$

b הוא פרמטר.

עבור $b = 3$ תחום הפתרונות האפשריים מוצג באיור I.

עבור $b = 6$ תחום הפתרונות האפשריים מוצג באיור II.

(שים לב: המשך השאלה בעמוד הבא.)



- ב. עבור $b = 3$, מהו הפתרון האופטימלי של הבעיה הנתונה? נמק את קביעתך.
- ג. עבור $b = 6$, הפתרון האופטימלי של הבעיה הנתונה הוא $(1, 4)$. הסבר מדוע.
- ד. מצא שני ערכים של b (השונים מ-6) שבעבורם הפתרון האופטימלי של הבעיה הנתונה הוא $(1, 4)$. נמק את תשובתך.
- ה. הופכים את סימן האי־שוויון של האילוץ הראשון בבעיית התכנון הלינארי שלפני סעיף ב, והוא כעת: $2x_1 + x_2 \geq 6$.
- בעבור $b = 3$ צייר את תחום הפתרונות האפשריים.
- מהו הפתרון האופטימלי לבעיה שהתקבלה לאחר השינוי הזה? נמק את תשובתך.
- ו. הופכים את סימן האי־שוויון של שני האילוצים הראשונים בבעיית התכנון הלינארי שלפני סעיף ב.
- בעבור $b = 3$ האילוצים הם עתה:

$$2x_1 + x_2 \geq 6$$

$$-2x_1 + x_2 \geq 2$$

$$x_1 + x_2 \leq 3$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

האם יש פתרון אופטימלי לבעיה שהתקבלה לאחר השינוי הזה? אם כן — מצא אותו,

אם לא — הסבר מדוע אין פתרון.

10. בשאלה זו שישנה סעיפים, א-ו, שאין קשר ביניהם. ענה על כל הסעיפים.
- א. בבעיית תובלה שלושה מקורות ושלושה יעדים. העלויות ליחידה מכל מקור לכל יעד נתונות בטבלה שלפניך.

מקורות	יעדים			היצע
	1	2	3	
1	8	9	6	18
2	4	5	8	a
3	5	10	7	10
ביקוש	12	15	15	

קבע מה צריך להיות ההיצע a כדי לקבל טבלת עלויות וביקושים, ללא יעד דמה וללא מקור דמה.

- ב. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית התובלה: $x_{11} = 9$, $x_{12} = 1$.

מקורות	יעדים			היצע
	1	2	3	
1	10 9	15 1	17	10
2	10	18	14	12
3	15	20	18	9
ביקוש	9	12	10	

העתק למחברתך את הטבלה, והשלם בה את הפתרון הבסיסי האפשרי לפי שיטת הפינה הצפונית-מערבית.

(שים לב: המשך השאלה בעמוד הבא.)

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

מקורות	יעדים			היצע	u_i
	1	2	3		
1	2 20	5	7	20	2
2	0	8 10	4	10	0
3	10	0	8 10	15	-8
ביקוש	20	15	10		
v_j	0	8	16		

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

(שים לב: המשך השאלה בעמוד הבא.)

/המשך בעמוד 19/

ד. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתון הערך של u_1 .

מקורות	יעדים			היצע	u_i
	1	2	3		
1	6	7 15	9 3	18	0
2	2 10	0 0	6	10	
3	7	12	10 10	10	
ביקוש	10	15	13		
v_j					

(1) העתק למחברתך את הטבלה, והשלם בה את הערכים של u_2, u_3, v_1, v_2, v_3 .

(2) הסבר מדוע הפתרון הנתון אינו פתרון אופטימלי.

(שים לב: המשך השאלה בעמוד הבא.)

/המשך בעמוד 20/

ה. בטבלה שלפניך נתון פתרון אפשרי לבעיית תובלה, ונתונים הערכים של

$u_1, u_2, u_3, v_1, v_2, v_3$ שמתאימים לפתרון זה.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	10 20	25	30	20	10
2	10 30	22	14 20	50	10
3	16	20 40	20 20	60	16
ביקוש	50	40	40		
v_j	0	4	4		

בחר בהיגד הנכון מבין ההיגדים 1-4 שלפניך. העתק אותו למחברתך ונמק את בחירתך.

1. הפתרון הנתון אינו פתרון בסיסי אפשרי.
2. הפתרון הנתון הוא פתרון בסיסי אפשרי אך לא אופטימלי.
3. הפתרון הנתון הוא פתרון אופטימלי יחיד.
4. הפתרון הנתון הוא פתרון אופטימלי, אך איננו פתרון אופטימלי יחיד.

(שים לב: המשך השאלה בעמוד הבא.)

ו. בטבלה שלפניך נתונה בעיית תובלה.

נתוני הבעיה, כולל העלויות ליחידה מכל מקור לכל יעד, מוצגים בטבלה שלפניך.

מקורות	יעדים			היצע
	1	2	3	
1	22	30	32	20
2	30	18	24	20
3	15	10	8	30
4	25	15	22	30
ביקוש	30	40	25	

נתון תנאי, שעל פיו אסור שיישאר מלאי במקור 1.

הבעיה הנתונה אינה מאוזנת, לכן הציע תלמיד להוסיף יעד דמה 4 אשר יקבל את

חמשת המוצרים מההיצע העודף.

העלויות ליחידה מכל מקור לכל יעד נתונות בטבלה שלפניך.

ערכי העלויות של יעד הדמה 4 הנתונים בטבלה הם הפרמטרים x, y, z, w .

מקורות	יעדים				היצע
	1	2	3	4	
1	22	30	32	x	20
2	30	18	24	y	20
3	15	10	8	z	30
4	25	15	22	w	30
ביקוש	30	40	25	5	

(שים לב: המשך הסעיף בעמוד הבא.)

מה צריך להיות הערך של כל אחד מן הפרמטרים x, y, z, w כדי שיהיה אפשר למצוא פתרון בסיסי אפשרי לבעיית התובלה הנתונה?

בחר בתשובה הנכונה מבין ארבע התשובות שלפניך, העתק אותה למחברתך, ונמק את בחירתך.

(M הוא מספר גדול מאוד.)

1. $x = 0, y = 0, z = 0, w = 0$

2. $x = M, y = M, z = M, w = M$

3. $x = M, y = 0, z = 0, w = 0$

4. $x = 0, y = M, z = M, w = M$

/המשך בעמוד 23/

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

א. $G = (V, E)$ הוא גרף מכוון המיוצג על ידי מטריצת הסמיכויות שלפניך.

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

(1) סרטט את הגרף G המיוצג על ידי מטריצת הסמיכויות.

(2) מצא את רכיב/רכיבי הקשירות החזקה (רק"חים — Strong Connected Components) שבגרף הנתון.

בעבור כל רק"ח שמצאת רשום את קבוצת הקדקודים שלו.

(3) מהו המספר המינימלי של קשתות שיש להוסיף לגרף הנתון כדי שהגרף יכלי רק"ח אחד בלבד? מהי הקשת או מהן הקשתות שיש להוסיף?

(4) כמה רכיבי קשירות חזקה (רק"חים) יש בגרף המוחלף $G^T = (V, E^T)$ של הגרף הנתון G , ומה הם?

שים לב — קשת מכוונת כלשהי (a, b) בגרף G נהפכת בגרף המוחלף G^T

לקשת מכוונת (b, a) .

ב. (אין קשר לסעיף א.)

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

a	→	b	→	c	→			
b	→	a	→	c	→	d	→	
c	→	e	→	a	→	b	→	
d	→	b	→					
e	→	c	→					

(1) סרטט את הגרף G המיוצג על ידי רשימת הסמיכויות.

(2) כמה רכיבי קשירות (Connected Components) יש בגרף G , ומה הם?

(3) הפעל אלגוריתם סריקה לעומק (DFS) על הגרף G החל בקדקוד a .

סרטט במחברתך רק את העץ הפורש (DFS) / היער הפורש (DFS) שמתקבל.

התבסס על הייצוג הנתון ברשימת הסמיכויות.

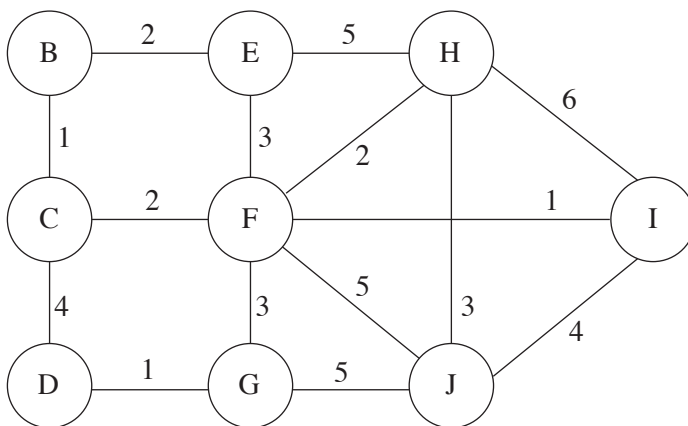
(4) הפעל אלגוריתם סריקה לרוחב (BFS) על הגרף G החל בקדקוד a .

סרטט במחברתך רק את העץ הפורש (BFS) / היער הפורש (BFS) שמתקבל.

התבסס על הייצוג הנתון ברשימת הסמיכויות.

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

א. לפניך רשת $G = (V, E)$.



(1) מצא את כל המסלולים הקצרים ביותר מקדקוד C לקדקוד J ברשת הנתונה.

תאר כל אחד מן המסלולים תיאור סכמתי.

(2) מצא את העץ הפורש המינימלי לפי האלגוריתם של קרוסקל. תאר את העץ הזה תיאור

סכמתי.

ב. נתונים גרף קשיר לא מכוון $G = (V, E)$ וכן פונקציית משקל $w: E \rightarrow \mathbb{R}^+$.

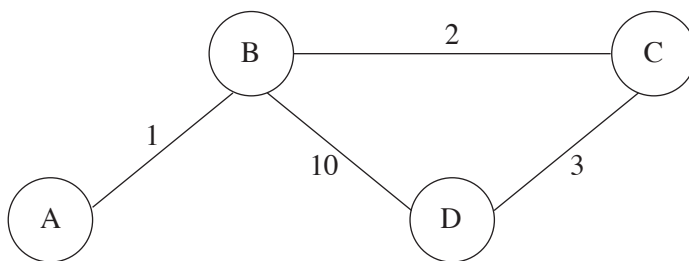
נגדיר פונקציית משקל חדשה $cl: E \rightarrow \mathbb{R}^+$ באופן הזה: לכל קשת $e \in E$, $cl(e) = w(e) + a$, a הוא מספר כלשהו, גדול מ-0.

תלמיד טען שאם T הוא עץ פורש מינימלי של G על פי האלגוריתם של קרוסקל תחת פונקציית המשקל w , אזי T הוא עץ פורש מינימלי של G על פי האלגוריתם של קרוסקל גם תחת פונקציית המשקל cl .

האם התלמיד צודק? אם כן — נמק מדוע, אם לא — הבא דוגמה שתפריך את טענת התלמיד.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. נתונים גרף קשיר לא מכוון $G = (V, E)$ וכן פונקציית משקל $w: E \rightarrow \mathbb{R}^+$.



בעבור הגרף G הנתון, עץ המסלולים הקצרים ביותר מקדקוד A לשאר קדקודי הגרף הוא:

$$A \rightarrow B \rightarrow C \rightarrow D$$

נגדיר פונקציית משקל חדשה $c_2: E \rightarrow \mathbb{R}^+$ באופן הזה: לכל קשת $e \in E$, $c_2(e) = w(e) + a$,

a הוא מספר כלשהו, גדול מ-0.

(1) הבא דוגמה ל- a שבעבורו עץ המסלולים הקצרים ביותר מקדקוד A לשאר

קדקודי הגרף לא ישתנה.

(2) הבא דוגמה ל- a שבעבורו עץ המסלולים הקצרים ביותר מקדקוד A לשאר

קדקודי הגרף ישתנה.

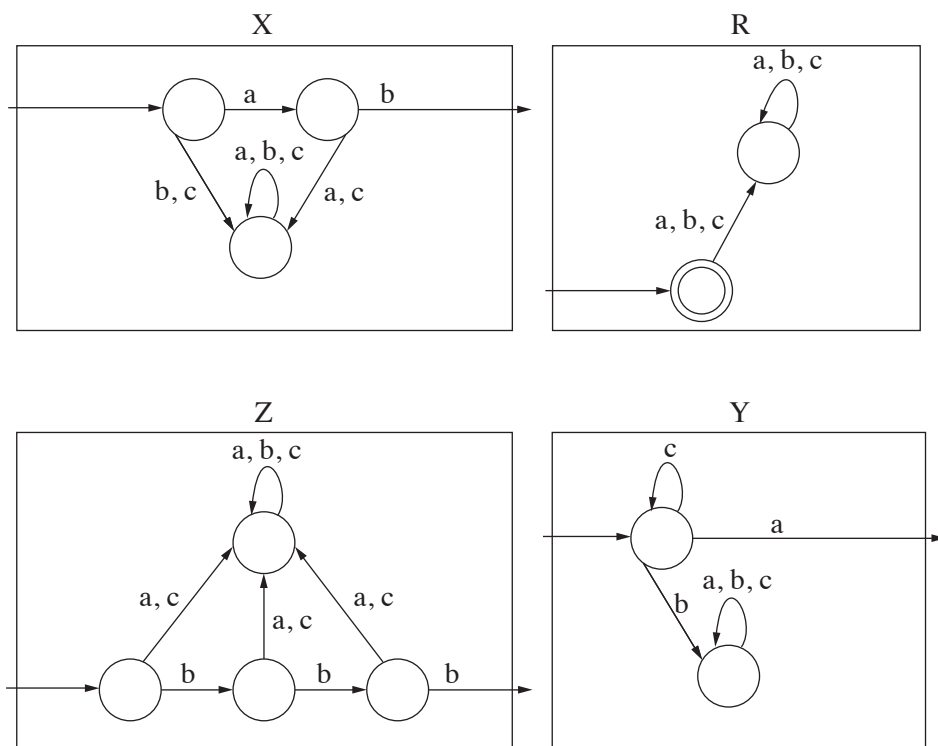
/המשך בעמוד 26/

מודלים חישוביים

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

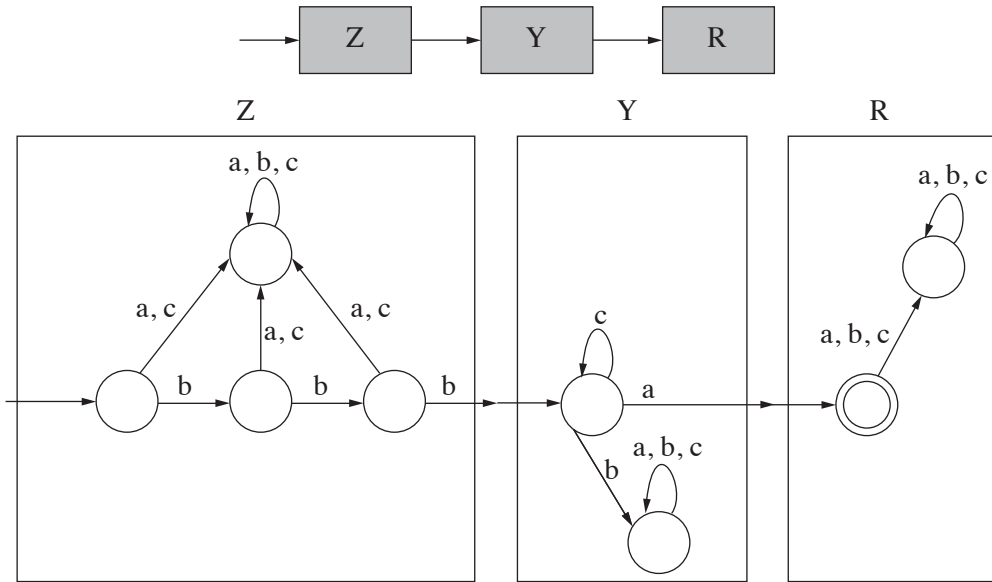
13. במפעל "אוטומטים בע"מ" מייצרים את ארבעת הסוגים של רכיבים לבניית אוטומטים סופיים דטרמיניסטיים שלפניך.

כל אחד מן הרכיבים אינו אוטומט, אך אפשר להרכיב מהם אוטומט סופי דטרמיניסטי, על ידי חיבורם זה לזה בסדר מסוים, משמאל לימין. אפשר להשתמש ביותר מרכיב אחד מכל סוג.

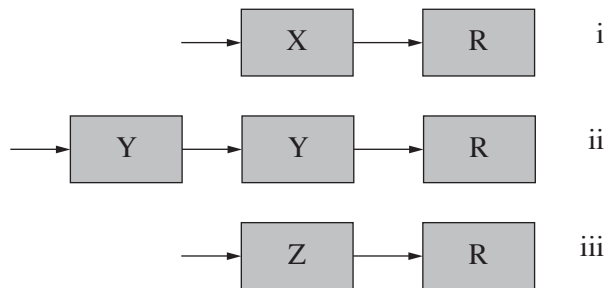


(שים לב: המשך השאלה בעמוד הבא.)

למשל כדי לבנות אוטומט שמקבל את השפה $\{b^3 c^n a \mid n \geq 0\}$ יש לחבר את הרכיבים שלפניך, בסדר המתואר, משמאל לימין:



א. כתוב את השפה שמקבל כל אחד מן האוטומטים i - iii שלפניך.



ב. נתונות השפות L_1, L_2, L_3 :

$$L_1 = \{ab^4 c^n a \mid n \geq 0\}$$

$$L_2 = \{c^n a^2 b^7 ab \mid n \geq 0\}$$

$$L_3 = \{b^6 c^n a^2 b^4 \mid n \geq 0\}$$

לכל אחת מן השפות L_1, L_2, L_3 רשום את הרכיבים הנחוצים מבין הרכיבים R, Z, Y, X משמאל לימין, לפי הסדר שצריך לחבר כדי לקבל אוטומט סופי דטרמיניסטי שיקבל את השפה. אפשר להשתמש ביותר מרכיב אחד מאותו סוג.

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

א. לפניך השפה L מעל הא"ב $\{a, b\}$.

L היא אוסף המילים שבכל אחת מהן הערך המוחלט של ההפרש בין מספר הפעמים שמופיע התו a לבין מספר הפעמים שמופיע התו b שווה ל-1, ובעבור כל רישא* של כל מילה השייכת לשפה L הערך המוחלט של ההפרש הזה הוא לכל היותר 1.

* רישא של מילה w היא כל מילה המתקבלת על ידי הורדת מספר כלשהו של תווים מסוף המילה w , כולל המילה הריקה, והמילה w עצמה.

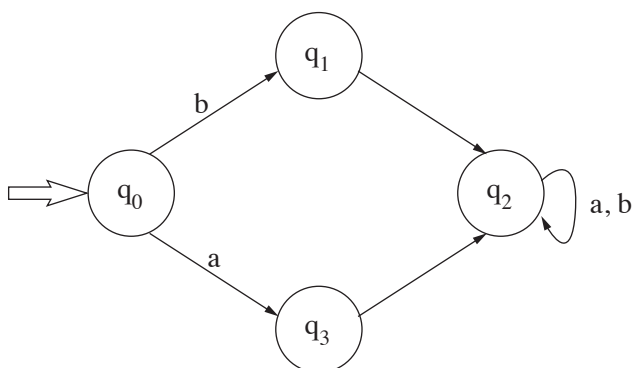
לדוגמה: $w = abba$

כל הרישות של המילה w הן: $\epsilon, a, ab, abb, abba$

דוגמה למילה ששייכת לשפה L : $ababa$

דוגמאות למילים שאינן שייכות לשפה L : $\epsilon, baab, aaabb$.

לפניך סרטוט חלקי של אוטומט סופי דטרמיניסטי המקבל את השפה L . בסרטוט חסרים מעברים, סימני קלט ומצבים מקבלים.



הסרטוט כולל את כל המצבים של האוטומט.

העתק למחברתך את הסרטוט, והשלם אותו כך שהאוטומט יקבל את השפה L . עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים, ולסמן את כל המצבים המקבלים.

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

(שים לב: המשך השאלה בעמוד הבא.)

ב. נתונות שתי שפות רגולריות L_1, L_2 .

$$L_3 = \{w_1 w_2 \mid w_1 \in L_1, w_2 \in L_2, |w_1| = |w_2|\}$$

הבא דוגמה לשתי שפות L_1, L_2 כך ש: L_1, L_2 הן שפות רגולריות

ו- L_3 היא שפה לא רגולרית. הסבר את תשובתך.

ג. לפניך השפות הרגולריות L_1, L_2, L_3 מעל הא"ב $\{a, b, c\}$.

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

$$L_2 = \{c^n \mid n \geq 0\}$$

$$L_3 = \{\varepsilon\}$$

לפניך השפה L .

$$L = \{c^n b^k a^j \mid n > 0, k, j \geq 0\}$$

הראה שהשפה L רגולרית באמצעות השפות L_1, L_2, L_3 ותכונות סגירות בלבד.

הסבר את תשובתך.

15. לפניך השפה L .

$$L = \left\{ a^s b^{2s} a^{i_1} b^{j_1} a^{i_2} b^{j_2} \dots a^{i_n} b^{j_n} \mid \begin{array}{l} s \geq 1, n \geq 1 \\ \text{לכל } k, 1 \leq k \leq n \text{ מתקיים } i_k \geq 1, j_k \geq 1 \end{array} \right\}$$

א. כתוב את המילה הקצרה ביותר בשפה L .

ב. בנה אוטומט מחסנית שיקבל את השפה L .

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

א. לפניך השפות L_1 ו- L_2 מעל הא"ב $\{0,1\}$:

$$L_1 = \{w \in \{0,1\}^* \mid \#_0(w) \leq \#_1(w) \leq 2 \times \#_0(w)\}$$

$$L_2 = \{0^s \mid s \geq 0\}$$

$\{0,1\}^*$ הוא אוסף כל המילים מעל הא"ב $\{0,1\}$, כולל המילה הריקה,

$\#_0(w)$ מציין את מספר ה-0ים במילה w ,

$\#_1(w)$ מציין את מספר ה-1ים במילה w .

נתונות שתי מילים ב- L_2 : $w_1 = 0^i$ ו- $w_2 = 0^j$ כך ש: $i < j$.

מצא מילה w כך שיתקיים:

$$w_1 \cdot w \in L_1$$

$$w_2 \cdot w \notin L_1$$

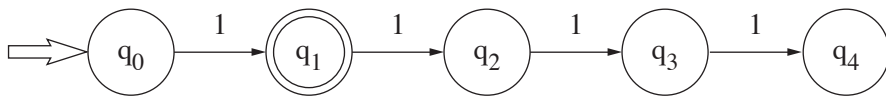
ב. (אין קשר לסעיף א.)

לפניך השפה L מעל הא"ב $\{0,1\}$.

$$L = \{10^i(11)^j \mid i, j \geq 0\} \cup \{10^k \mid k \geq 0\}$$

לפניך סרטוט חלקי של אוטומט סופי לא דטרמיניסטי המקבל את השפה L .

בסרטוט חסרים מעברים, סימני קלט ומצבים מקבלים.



הסרטוט כולל את כל המצבים של האוטומט הלא דטרמיניסטי.

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

עליך להשלים את המעברים החסרים ואת סימני הקלט החסרים, ולסמן את המצבים

המקבלים החסרים.

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

תכנות מונחה עצמים

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

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

א. לפניך קטע קוד בפעולה ראשית במחלקה Run:

```
C c = new A();
B b1 = (B) (new A());
B b2 = new D();
A a = new D();
```

(1) הנח כי A, B, C, D הן מחלקות, ונתון כי לכל אחת מהן יש מחלקה יורשת אחת לכל היותר.

כתוב שתי אפשרויות להיררכיית המחלקות כך שקטע הקוד יהיה תקין.

(2) נתון גם כי הפעולה f() מוגדרת במחלקה B בלבד, וכן נתון כי בתכנית הראשית נכללות גם ההוראות האלה:

```
c.f();
b1.f();
```

קבע באיזו מבין שתי האפשרויות להיררכיית המחלקות שכתבת בתת-סעיף א (1) ההוראות האלה תקינות.

העתק את היררכיית המחלקות הזו, והסבר מדוע בהיררכייה האחרת ההוראות אינן תקינות.

ב. לפניך קטע תכנית בפעולה הראשית במחלקה Program:

```
CC c = new AA();
BB b1 = (BB) (new AA(2));
```

קבע מי מבין AA, BB, CC לא יכול להיות ממשק. הסבר מדוע.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. לפניך המחלקות A , B והמחלקה Run ובה הפעולה הראשית main . רשום את הפלט של הפעולה הראשית.

```
public class A
{
    public A()
    {
        f();
    }
    public void f()
    {
        System.out.println("Class A");
    }
}

public class B extends A
{
    public B()
    {
        super();
    }
    public void f()
    {
        System.out.println("Class B");
    }
}

public Class Run
{
    public static void main(String[] args)
    {
        A a = new B();
    }
}
```


18. לפניך המחלקות AA, BB, CC, DD והמחלקה Run ובה פעולה ראשית.

```
public class AA
{
    protected int i;
    public AA(int i)
    {
        this.i = i;
    }
}
```

```
public class BB extends AA
{
    public BB(int i)
    {
        super(i + 1);
    }
    public boolean what(Object other)
    {
        return ((other != null) &&
                (other instanceof BB) &&
                (this.i == ((BB)other).i));
    }
    public boolean what (BB other, int k)
    {
        return ((other != null) &&
                (this.i - k == ((BB)other).i));
    }
}
```

```
public class CC extends BB
{
    public CC(int i)
    {
        super(i);
    }
}
```

```
public class DD extends BB
{
    public DD(int i)
    {
        super(i + 1);
    }
    public boolean what(BB other , int k)
    {
        return super.what(other , 1);
    }
}

public class Run
{
    public static void main(String[] args)
    {
        AA a = new AA(1);
        BB b = new BB(1);
        CC c = new CC(1);
        DD d = new DD(1);
        BB b1 = new DD(1);
        AA c1 = new CC(1);
        /***/
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

א. סרטט את היררכיית המחלקות AA, BB, CC, DD.

ב. עקוב אחר הפעולה הראשית.

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

ג. לפניך הוראות iii-i.

- i a = c;
- ii b = a;
- iii c = (CC) b1;

בעבור כל אחת מן ההוראות iii-i בצע:

— כתוב את ההוראה במקום /***/ בפעולה הראשית של המחלקה Run.

— קבע אם ההוראה תקינה או אינה תקינה.

אם ההוראה תקינה — הצג את השינויים שנוצרו בעצמים בעקבות ההוראה.

אם ההוראה אינה תקינה — נמק את קביעתך, וכתוב אם השגיאה היא שגיאת הידור (קומפילציה) או שגיאת זמן ריצה.

ד. לפניך הוראות iv-i.

- i System.out.println(b1.what(b));
- ii System.out.println(b1.what(b , 1));
- iii System.out.println(((CC) c1).what(c));
- iv System.out.println(d.what(a));

בעבור כל אחת מן ההוראות iv-i בצע:

— כתוב את ההוראה במקום /***/ בפעולה הראשית של המחלקה Run.

— כתוב מה יהיה הפלט בעקבות כל אחת מן ההוראות.

19. לפניך הממשק `IThing`, המחלקות `A`, `B`, `C`, `D`, ופעולה ראשית במחלקה `Run`.

א. עקוב אחר הפעולה הראשית, וכתוב את הפלט המתקבל.

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

```
public interface IThing
{
    public int value();
}
```

```
public class A implements IThing
{
    private int a;
    public A(int a)
    {
        this.a = a;
    }
    public int value()
    {
        return this.a;
    }
}
```

```
public class B implements IThing
{
    protected IThing a , b;
    public B(IThing a , IThing b)
    {
        this.a = a;
        this.b = b;
    }
    public int value()
    {
        return this.a.value() + this.b.value();
    }
}
```

(שים לב: המשך הסעיף בעמוד הבא.)

```
public class C extends B
{
    public C(IThing a, IThing b)
    {
        super(a , b);
    }
    public int value()
    {
        return this.a.value() * this.b.value();
    }
}
```

```
public class D implements IThing
{
    private IThing[] things;
    private int limit;
    public D(IThing[] things , int limit)
    {
        this.things = things;
        this.limit = limit;
    }
    public int value()
    {
        int val = 0;
        for (int i = 0; i < limit; i++)
        {
            val += this.things[i].value();
        }
        return val;
    }
}
```

(שים לב: המשך הסעיף בעמוד הבא.)

```
public class Run
{
    public static void main(String[] args)
    {
        IThing[] things = new IThing[5];
        things[0] = new A(2);
        things[1] = new B(things[0] , things[0]);
        things[2] = new C(things[0] , things[1]);
        things[3] = things[2];
        things[4] = new D(things , 4);    // (*)
        for(int i = 0; i < things.length; i++)
        {
            System.out.println(things[i].value());
        }
    }
}
```

ב. הסבר בקצרה מה תהיה ההשפעה על ריצת התכנית אם בשורה המסומנת ב- (*) יחליפו את ההוראה הקיימת בהוראה שלפניך:

```
things[4] = new D(things , 5);
```

/המשך בעמוד 38/

20. החברה "צעצועים זה הם" מעוניינת למחשב את מאגר הצעצועים במחסן המרכזי שלה.

כדי לעשות זאת הוגדרו שתי מחלקות בעבור שני סוגים של צעצועים:

המחלקה **Doll** בעבור בובות, והמחלקה **Car** בעבור מכוניות צעצוע.

במחלקה **Doll** חמש תכונות ושתי פעולות.

התכונות: שם הבובה, מחיר הבסיס של הבובה, צבע שער, מספר האביזרים המצורפים, מחיר לאביזר (לכל האביזרים מחיר זהה).

הפעולות:

(i) החזרת מחיר הבובה לצרכן. מחיר הבובה לצרכן הוא מחיר הבסיס שלה, בתוספת מחיר

האביזרים המצורפים אליה (המחושב על פי מספר האביזרים כפול המחיר לאביזר).

(ii) עדכון מחיר הבסיס של בובה על ידי הגדלתו ב- p אחוזים. p הוא מספר ממשי המתקבל

כפרמטר.

במחלקה **Car** ארבע תכונות ושתי פעולות.

התכונות: שם מכונית הצעצוע, מחיר הבסיס של המכונית, סיווג גודל המכונית, צבע המכונית.

התכונה "סיווג גודל המכונית" מיוצגת על ידי מספר שלם: 0 – מכונית קטנה;

1 – מכונית בינונית; 2 – מכונית גדולה.

הפעולות:

(i) החזרת מחיר המכונית לצרכן. מחיר המכונית לצרכן נקבע על פי גודלה.

המחיר לצרכן של מכונית קטנה הוא מחיר הבסיס,

המחיר לצרכן של מכונית בינונית הוא מחיר בסיס + 15 ש"ח,

והמחיר לצרכן של מכונית גדולה הוא מחיר הבסיס + 30 ש"ח.

(ii) עדכון מחיר הבסיס של מכונית על ידי הגדלתו ב- p אחוזים. p הוא מספר ממשי המתקבל

כפרמטר.

(שים לב: המשך השאלה בעמוד הבא.)

לפניך תרשים המחלקות של מאגר הצעצועים הממוחשב:

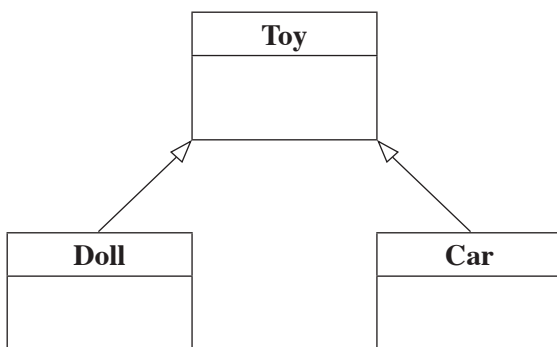
Doll
<ul style="list-style-type: none"> — String name — double basePrice — String color — int accNums — double accPrice
<ul style="list-style-type: none"> + double price() + void updatePrice(double percent) + Doll(String name , double basePrice , String color , int accNum , double accPrice)

Car
<ul style="list-style-type: none"> — String name — double basePrice — String color — int size
<ul style="list-style-type: none"> + double price() + void updatePrice(double percent) + Car (String name , double basePrice , String color , int size)

מקרא:
— מציין private
+ מציין public

(שים לב: סעיפי השאלה בעמוד הבא.)

- א. הוגדרה מחלקה חדשה **Toy**, שהמחלקות **Doll** ו-**Car** יורשות ממנה. לפניך שלד של תרשים מחלקות בעבור המחלקות **Toy**, **Doll** ו-**Car**. העתק את התרשים למחברתך וכתוב בו לכל מחלקה את התכונות ואת כותרות הפעולות, באופן המתאים ביותר לעקרונות תכנות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism). כתוב תיעוד לכל תכונה.



- בסעיפים ב-ג שלפניך התייחס לתרשים המחלקות שכתבת בסעיף א.
- ב. ממש ב-Java את הפעולה price, בכל המחלקות שהיא מופיעה בהן.
- ג. כתוב ב-Java פעולה ראשית שתבצע את שתי המשימות (1) ו-(2) שלפניך.
- (1) יצירת עצם אחד מטיפוס **Doll**, ועצם אחד מטיפוס **Car** (עם ערכים כרצונך).
- (2) חישוב והדפסה של סכום המחירים לצרכן של שני הצעצועים (בובה ומכונית).

תכנות מונחה עצמים

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

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

א. לפניך קטע קוד בפעולה ראשית במחלקה Run:

```
C c = new A();
```

```
B b1 = (B) (new A());
```

```
B b2 = new D();
```

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

(1) הנח כי A, B, C, D הן מחלקות ונתון כי לכל אחת מהן מחלקה יורשת אחת
לכל היותר.

כתוב שתי אפשרויות להיררכיית המחלקות כך שקטע הקוד יהיה תקין.

(2) נתון גם כי הפעולה F() מוגדרת במחלקה B בלבד, כמו כן נתון כי בתכנית הראשית
נכללות גם ההוראות האלה:

```
c.F();
```

```
b1.F();
```

קבע באיזו מבין שתי האפשרויות להיררכיית המחלקות שכתבת בתת־סעיף א (1)
ההוראות האלה תקינות.

העתק את היררכיית המחלקות הזו, והסבר מדוע בהיררכייה האחרת ההוראות
אינן תקינות.

ב. לפניך קטע תכנית בפעולה ראשית במחלקה Program:

```
CC c = new AA();
```

```
BB b1 = (BB) (new AA(2));
```

קבע מי מבין AA, BB, CC לא יכול להיות ממשק. הסבר מדוע.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

ג. לפניך המחלקות A, B והמחלקה Run ובה הפעולה הראשית Main. רשום את הפלט של הפעולה הראשית.

```
public class A
{
    public A()
    {
        F();
    }
    public virtual void F()
    {
        Console.WriteLine("Class A");
    }
}
```

```
public class B : A
{
    public B(): base() { }
    public override void F()
    {
        Console.WriteLine("Class B");
    }
}
```

```
public Class Run
{
    public static void Main()
    {
        A a = new B();
    }
}
```

22. לפניך המחלקות AA, BB, CC, DD והמחלקה Run ובה פעולה ראשית.

<pre> public class AA { protected int i; public AA(int i) { this.i = i; } } public class BB : AA { public BB(int i) : base(i + 1) { } public virtual bool What(Object other) { return ((other != null) && (other is BB) && (this.i == ((BB)other).i)); } public virtual bool What (BB other, int k) { return ((other != null) && (this.i - k == ((BB)other).i)); } } public class CC : BB { public CC(int i) : base (i) { } } </pre>	<pre> public class DD : BB { public DD(int i) : base(i + 1){ } public override bool What (BB other , int k) { return base.What(other , 1); } } public class Run { public static void Main() { AA a = new AA(1); BB b = new BB(1); CC c = new CC(1); DD d = new DD(1); BB b1 = new DD(1); AA c1 = new CC(1); /*** } } </pre>
--	--

(שים לב: המשך השאלה בעמוד הבא.)

א. סרטט את היררכיית המחלקות AA, BB, CC, DD.

ב. עקוב אחר הפעולה הראשית.

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

ג. לפניך הוראות iii-i.

- i a = c;
- ii b = a;
- iii c = (CC) b1;

בעבור כל אחת מן ההוראות iii-i בצע:

— כתוב את ההוראה במקום /***/ בפעולה הראשית של המחלקה Run.

— קבע אם ההוראה תקינה או אינה תקינה.

אם ההוראה תקינה — הצג את השינויים שנוצרו בעצמים בעקבות ההוראה.

אם ההוראה אינה תקינה — נמק את קביעתך, וכתוב אם השגיאה היא שגיאת הידור

(קומפילציה) או שגיאת זמן ריצה.

ד. לפניך הוראות iv-i.

- i Console.WriteLine(b1.What(b));
- ii Console.WriteLine(b1.What(b , 1));
- iii Console.WriteLine(((CC) c1).What(c));
- iv Console.WriteLine(d.What(a));

בעבור כל אחת מן ההוראות iv-i בצע:

— כתוב את ההוראה במקום /***/ בפעולה הראשית של המחלקה Run.

— כתוב מה יהיה הפלט בעקבות כל אחת מן ההוראות.

23. לפניך הממשק `IThing`, המחלקות `A`, `B`, `C`, `D`, ופעולה ראשית במחלקה `Run`.

א. עקוב אחר הפעולה הראשית, וכתוב את הפלט המתקבל.

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

```
public interface IThing
{
    int Value();
}
```

```
public class A : IThing
{
    private int a;
    public A(int a)
    {
        this.a = a;
    }
    public int Value()
    {
        return this.a;
    }
}
```

```
public class B : IThing
{
    protected IThing a , b;
    public B(IThing a , IThing b)
    {
        this.a = a;
        this.b = b;
    }
    public virtual int Value()
    {
        return this.a.Value() + this.b.Value();
    }
}
```

(שים לב: המשך הסעיף בעמוד הבא.)

```
public class C : B
{
    public C(IThing a, IThing b) : base(a , b) { }
    public override int Value()
    {
        return this.a.Value() * this.b.Value();
    }
}
```

```
public class D : IThing
{
    private IThing[] things;
    private int limit;
    public D (IThing[] things , int limit)
    {
        this.things = things;
        this.limit = limit;
    }
    public int Value()
    {
        int val = 0;
        for (int i = 0; i < limit; i++)
        {
            val += this.things[i].Value();
        }
        return val;
    }
}
```

(שים לב: המשך הסעיף בעמוד הבא.)

```
public class Run
{
    public static void Main()
    {
        IThing[] things = new IThing[5];
        things[0] = new A(2);
        things[1] = new B(things[0] , things[0]);
        things[2] = new C(things[0] , things[1]);
        things[3] = things[2];
        things[4] = new D(things , 4);    // (*)
        for(int i = 0; i < things.Length; i++)
        {
            Console.WriteLine(things[i].Value());
        }
    }
}
```

ב. הסבר בקצרה מה תהיה ההשפעה על ריצת התכנית אם בשורה המסומנת ב- (*) יחליפו את ההוראה הקיימת בהוראה שלפניך:

```
things[4] = new D(things , 5);
```

24.

החברה "צעצועים זה הם" מעוניינת למחשב את מאגר הצעצועים במחסן המרכזי שלה.

כדי לעשות זאת הוגדרו שתי מחלקות בעבור שני סוגים של צעצועים:

המחלקה **Doll** בעבור בובות, והמחלקה **Car** בעבור מכוניות צעצוע.

במחלקה **Doll** חמש תכונות ושתי פעולות.

התכונות: שם הבובה, מחיר בסיס של הבובה, צבע שער, מספר האביזרים המצורפים, מחיר לאביזר (לכל האבזרים מחיר זהה).

הפעולות:

(i) החזרת מחיר הבובה לצרכן. מחיר הבובה לצרכן הוא מחיר הבסיס שלה, בתוספת מחיר

האביזרים המצורפים אליה (המחושב על פי מספר האביזרים כפול המחיר לאביזר).

(ii) עדכון מחיר הבסיס של בובה על ידי הגדלתו ב- p אחוזים. p הוא מספר ממשי המתקבל

כפרמטר.

במחלקה **Car** ארבע תכונות ושתי פעולות.

התכונות: שם מכונית הצעצוע, מחיר הבסיס של המכונית, סיווג גודל המכונית, צבע המכונית.

התכונה "סיווג גודל המכונית" מיוצגת על-ידי מספר שלם: 0 — מכונית קטנה;

1 — מכונית בינונית; 2 — מכונית גדולה.

הפעולות:

(i) החזרת מחיר המכונית לצרכן. מחיר המכונית לצרכן נקבע על פי גודלה.

המחיר לצרכן של מכונית קטנה הוא מחיר הבסיס,

המחיר לצרכן של מכונית בינונית הוא מחיר בסיס + 15 ש"ח,

והמחיר לצרכן של מכונית גדולה הוא מחיר הבסיס + 30 ש"ח.

(ii) עדכון מחיר הבסיס של מכונית על-ידי הגדלתו ב- p אחוזים. p הוא מספר ממשי

המתקבל כפרמטר.

(שים לב: המשך השאלה בעמוד הבא.)

לפניך תרשים המחלקות של מאגר הצעצועים הממוחשב:

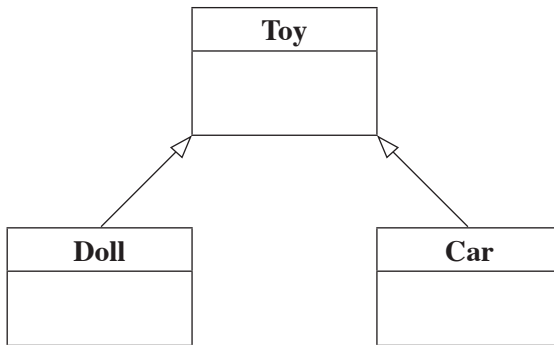
Doll
<ul style="list-style-type: none"> — string name — double basePrice — string color — int accNums — double accPrice
+ double Price() + void UpdatePrice(double percent) + Doll(string name , double basePrice , string color , int accNum , double accPrice)

Car
<ul style="list-style-type: none"> — string name — double basePrice — string color — int size
+ double Price() + void UpdatePrice(double percent) + Car (string name , double basePrice , string color , int size)

	מקרא:
—	מציין private
+	מציין public

(שים לב: סעיפי השאלה בעמוד הבא.)

- א. הוגדרה מחלקה חדשה **Toy** שהמחלקות **Doll** ו-**Car** יורשות ממנה. לפניך שלד של תרשים מחלקות בעבור המחלקות **Doll**, **Toy** ו-**Car**. העתק את התרשים למחברתך וכתוב בו לכל מחלקה את התכונות ואת כותרות הפעולות, באופן המתאים ביותר לעקרונות תכנות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism). כתוב תיעוד לכל תכונה.



- בסעיפים ב-ג שלפניך התייחס לתרשים המחלקות שכתבת בסעיף א.
- ב. ממש ב- C# את הפעולה Price, בכל המחלקות שהיא מופיעה בהן.
- ג. כתוב ב- C# פעולה ראשית שתבצע את שתי המשימות (1) ו- (2) שלפניך.
- (1) יצירת עצם אחד מטיפוס **Doll**, ועצם אחד מטיפוס **Car** (עם ערכים כרצונך).
- (2) חישוב והדפסה של סכום המחירים לצרכן של שני הצעצועים (בובה ומכונית).

בהצלחה!