

מדעי המחשב

הוראות

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.

פרק ראשון – 25×2 – 50 נקודות

פרק שני – 25×2 – 50 נקודות

סך הכול – 100 נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

ד. הוראות מיוחדות:

(1) רשמו על הכריכה החיצונית של המחברת את שם המסלול שלמדתם.

המסלול הוא אחד משלושת המסלולים האלה: אלגוריתמים, מודלים חישוביים, תכנות מונחה עצמים.

(2) את כל התוכניות שיש לכתוב בשפת מחשב בפרקים הראשון והשני כתבו בשפה אחת בלבד – Java או C#.

הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

בשאלון זה שני פרקים. יש לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים בשפת Java: בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה:

Scanner input = new Scanner (System.in);

שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

פרק ראשון (50 נקודות)

ענו על שתיים מן השאלות 1-3 (לכל שאלה - 25 נקודות).

1. נתונה המחלקה **Game** - משחק מחשב, ולה שתי תכונות:

- name - שם המשחק, מטיפוס מחרוזת.
 - price - מחיר המשחק - מספר הגדול מ-0, מטיפוס שלם.
- הניחו שיש פעולות get/Set ו-set/Set לתכונות המחלקה.

נתונה המחלקה **Store** - חנות משחקי מחשב, ולה תכונה אחת:

- lst - הפניה לשרשרת חוליות שאינה ריקה, מטיפוס **Game**. כל חוליה בשרשרת מכילה משחק הנמכר בחנות.

הערה: המשחקים אינם מסודרים בשרשרת בסדר מסוים, וכל משחק מופיע פעם אחת בלבד.

א. ממשו את הפעולה של ממשק המחלקה **Store** שלפניכם:

Java - public int remove (int n, int pr)

C# - public int Remove (int n, int pr)

הפעולה תמחק מן השרשרת n משחקים שמחיר כל אחד מהם pr. אם יש יותר מ-n משחקים שמחירם pr,

יימחקו רק n המשחקים הראשונים מביניהם. אם יש פחות מ-n משחקים שמחירם pr, רק הם יימחקו.

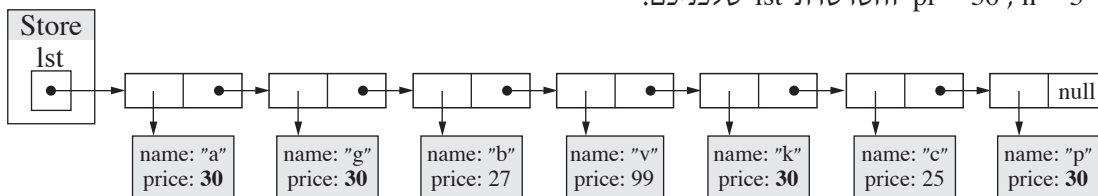
הפעולה תחזיר את כמות המשחקים שנמחקו (כלומר מקסימום n, אך ייתכן שפחות).

הניחו ש-n ו-pr גדולים מ-0.

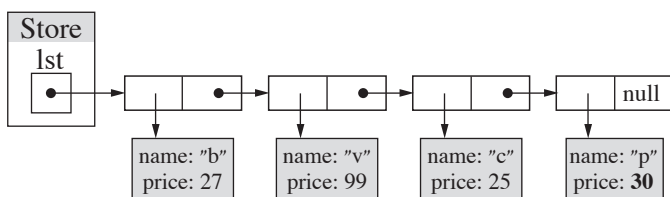
הערה: שאר המשחקים בשרשרת יישארו באותו הסדר. אם אין בשרשרת שום משחק במחיר pr, השרשרת תישאר

ללא שום שינוי והפעולה תחזיר 0.

דוגמה: בעבור n = 3, pr = 30 והשרשרת lst שלפניכם:



הפעולה תחזיר 3 והשרשרת תיראה כך בתום הפעולה:



הסבר: יש בשרשרת ארבעה משחקים שמחירם 30. מכיוון ש-n = 3, נמחקו שלושת המשחקים הראשונים בשרשרת

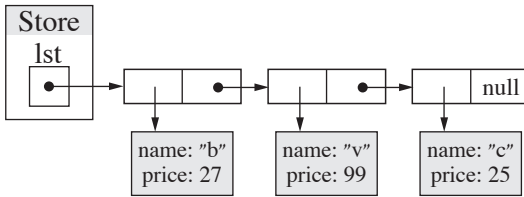
שמחירם 30, והפעולה החזירה 3.

/המשך בעמוד 3/

(שימו לב: המשך השאלה בעמוד הבא.)

דוגמה נוספת: בעבור אותה השרשרת מן הדוגמה המקורית שלעיל ו- $n = 5$, $pr = 30$, הפעולה תחזיר 4

והשרשרת תיראה כך בתום הפעולה:



הסבר: יש רק ארבעה משחקים שמחירים 30. לכן נמחקו מן השרשרת ארבעת המשחקים שמחירים 30, והפעולה החזירה 4.

ב. ממשו את הפעולה של ממשק המחלקה **Store** שלפניכם:

Java – `public int removeCheap (int num)`

C# – `public int RemoveCheap (int num)`

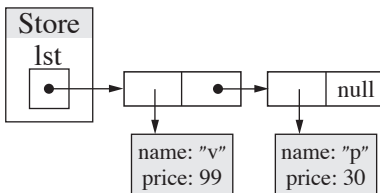
הפעולה תמחק מן השרשרת את num המשחקים הזולים ביותר. הפעולה תחזיר את סכום המחירים הכולל של כל המשחקים שנמחקו.

הניחו ש- num גדול מ-0 וקטן מכמות המשחקים בשרשרת.

הערות:

- המשחקים בשרשרת שלא נמחקו יישארו באותו הסדר.
- אם כמות המשחקים שמחירים זהה גדולה מכמות המשחקים שצריך למחוק מהם, אין חשיבות איזה מהם יימחק.
- אפשר להשתמש בפעולה שבסעיף א.

דוגמה: בעבור אותה השרשרת מן הדוגמה המקורית בסעיף א ו- $num = 5$, הפעולה תחזיר 142, והשרשרת תיראה כך:



הסבר: חמשת המשחקים הזולים יותר ($30+30+30+27+25$) נמחקו מן השרשרת וסכום מחירים הכולל הוא 142.

בשרשרת נשאר משחק אחד שמחירו 99 ואחד שמחירו 30 (אפשר להשאיר בשרשרת משחק אחר שעולה 30, אין חשיבות איזה מהם יישאר).

2. שימו לב: לשאלה זו שני נוסחים: בשפת Java בעמודים 4-5 ובשפת C# בעמודים 6-7.

לפותרים בשפת Java

א. לפניכם הפעולה mmm, המקבלת תור q ובו מספרים הגדולים מ-0, ומספר שלם z – הגדול מ-0.

```
public static boolean mmm (Queue<Integer> q, int z)
{
    q.insert (0);
    int num = q.head();
    int y = 0;
    while (q.head() > 0)
    {
        if (y < z)
        {
            if (q.head() == num)
            {
                y++;
            }
            else
            {
                num = q.head();
                y = 1;
            }
        }
        q.insert (q.remove());
    }
    q.remove();
    return y == z;
}
```

נתון תור q מטיפוס שלם:

ראש התור						סוף התור
	1	3	1	1	1	2

(1) עקבו בעזרת טבלת המעקב שלפניכם אחר הפעולה mmm(q, 4), וכתבו מה הפעולה מחזירה.

התור q	num	y	y < z	q.head() == num

(2) הסבירו מה הפעולה mmm עושה.

(3) מהי סיבוכיות זמן הריצה של הפעולה mmm? נמקו את תשובתכם.

(שימו לב: המשך השאלה בעמוד הבא.)

ב. לפניכם הפעולה what, המקבלת תור – q ובו מספרים הגדולים מ־ 0, ואת גודל התור – n.

```
public static int what (Queue<Integer> q, int n)
{
    if (mmm (q, n))
        return n;
    return what (q, n – 1);
}
```

נתון תור – q מטיפוס שלם:

ראש התור	סוף התור						
		2	1	1	1	3	1

- (1) עקבו אחר הפעולה what (q, 6), וכתבו מה הפעולה מחזירה (אין צורך לעקוב אחר הפעולה mmm).
 המעקב יכלול בכל קריאה את הערכים של q, n ואת הערך המוחזר.
 לפניכם הצעה לטבלת מעקב (אין חובה להשתמש בטבלה זו).

ערך מוחזר	mmm (q, n) == true	הערך n שמתקבל בפעולה	התור q שמתקבל בפעולה

- (2) הסבירו מה הפעולה what עושה.
 (3) מהי סיבוכיות זמן הריצה של הפעולה what ? נמקו את תשובתכם.

לפותרים בשפת C#

א. לפניכם הפעולה Mmm, המקבלת תור q ובו מספרים הגדולים מ־0, ומספר שלם z – הגדול מ־0.

```
public static bool Mmm (Queue<int> q, int z)
{
    q.Insert (0);
    int num = q.Head();
    int y = 0;
    while (q.Head() > 0)
    {
        if (y < z)
        {
            if (q.Head() == num)
            {
                y++;
            }
            else
            {
                num = q.Head();
                y = 1;
            }
        }
        q.Insert (q.Remove());
    }
    q.Remove();
    return y == z;
}
```

נתון תור q מטיפוס שלם:

ראש התור						סוף התור
	1	3	1	1	1	2

(1) עקבו בעזרת טבלת המעקב שלפניכם אחר הפעולה Mmm(q, 4), וכתבו מה הפעולה מחזירה.

q התור	num	y	y < z	q.Head() == num

(2) הסבירו מה הפעולה Mmm עושה.

(3) מהי סיבוכיות זמן הריצה של הפעולה Mmm? נמקו את תשובתכם.

(שימו לב: המשך השאלה בעמוד הבא.)

ב. לפניכם הפעולה What, המקבלת תור – q ובו מספרים הגדולים מ-0, ואת גודל התור – n.

```
public static int What (Queue<int> q, int n)
{
    if (Mmm (q, n))
        return n;
    return What (q, n – 1);
}
```

נתון תור – q מטיפוס שלם:

ראש התור	סוף התור						
		2	1	1	1	3	1

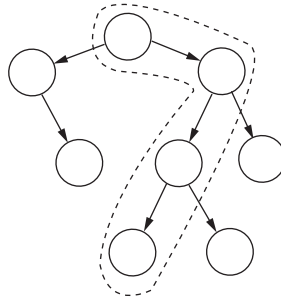
- (1) עקבו אחר הפעולה What (q, 6), וכתבו מה הפעולה מחזירה (אין צורך לעקוב אחר הפעולה Mmm).
 המעקב יכלול בכל קריאה את הערכים של q, n ואת הערך המוחזר.
 לפניכם הצעה לטבלת מעקב (אין חובה להשתמש בטבלה זו).

התור q שמתקבל בפעולה	הערך n שמתקבל בפעולה	Mmm (q, n) == true	ערך מוחזר

- (2) הסבירו מה הפעולה What עושה.
 (3) מהי סיבוכיות זמן הריצה של הפעולה What? נמקו את תשובתכם.

3. "עץ K-שמאלי" הוא עץ בינרי ובו כל מסלול משורש העץ לעלה כלשהו מכיל לכל היותר K קשתות הפונות שמאלה (כלומר מקסימום K בנים שמאליים).
שימו לב: ב"עץ K-שמאלי" ייתכן שאין שום מסלול המכיל K קשתות הפונות שמאלה (כלומר ייתכן שבכל אחד מן המסלולים יש פחות קשתות שמאליות מ-K).

דוגמה:



עץ זה אינו "עץ 1-שמאלי" כי בעץ יש מסלול שבו יותר מקשת אחת שמאלית (במסלול המקווקו יש שתי קשתות שמאליות). לעומת זאת, עץ זה הוא "עץ 2-שמאלי" כי אין בו מסלול שבו יותר משתי קשתות שמאליות (ובאותה מידה הוא גם "עץ 3-שמאלי", וגם "עץ 4-שמאלי" וכן הלאה).

- א. סרטטו "עץ 1-שמאלי" (כלומר שבכל מסלול של העץ יש קשת אחת שמאלית לכל היותר), ובו 8 צמתים ומקסימום 4 רמות.

- ב. כתבו פעולה חיצונית ששמה isLeftK בשפת Java או IsLeftK בשפת C#, המקבלת עץ בינרי – root מטיפוס שלם שאינו ריק, וערך K שאינו שלילי, ומחזירה true אם העץ הוא "עץ K-שמאלי", ואם לא – היא מחזירה false.

פרק שני (50 נקודות)

בפרק זה שאלות בשלושה מסלולים:

אלגוריתמים, עמודים 9–10.

מודלים חישוביים, עמודים 11–13.

תכנות מונחה עצמים (שפת Java ושפת C#), עמודים 14–25.

יש לענות על שתי שאלות במסלול שלמדתם (לכל שאלה – 25 נקודות).

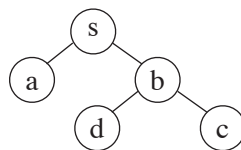
אלגוריתמים

4. בשאלה זו שני סעיפים, א–ב, שאין קשר ביניהם. ענו על שני הסעיפים.

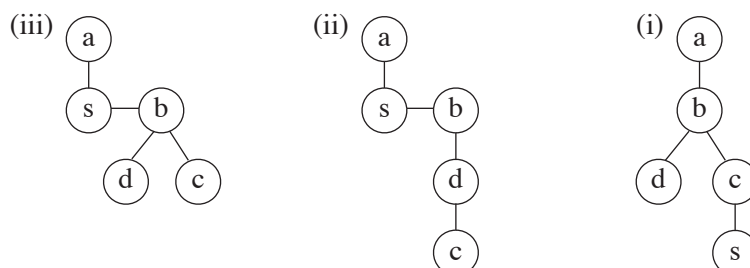
- א. לפניכם שש טענות 1–6. בחרו בארבע מהן, כתבו את מספר הטענה, וציינו בנוגע לכל טענה שבחרתם אם היא נכונה או לא נכונה. אם הטענה נכונה – נמקו מדוע, ואם היא אינה נכונה – הביאו דוגמה נגדית מגרף שיש בו 4 קודקודים לפחות.
- (1) נתון גרף לא מכוון ובו n קודקודים. אם יש בגרף $n-1$ קשתות, בהכרח אין בו מעגלים.
 - (2) נתון גרף ממושקל (אי-שלילי) ובו מעגל אחד לפחות. העץ הפורש המינימלי של הגרף בהכרח אינו מכיל את הקשת עם המשקל הגבוה ביותר בגרף.
 - (3) נתון גרף לא מכוון ובו n קודקודים ורכיב קשירות אחד. ייתכן שלאחר מחיקת קודקוד אחד (והקשתות שמחוברות אליו), יהיו בגרף $n-1$ רכיבי קשירות.
 - (4) נתון גרף לא מכוון ללא מעגלים. סריקת DFS תמצא תמיד את המרחק המינימלי בין שני קודקודים בגרף שיש ביניהם מסלול.
 - (5) נתון גרף מכוון ללא מעגלים. סריקת DFS תמצא תמיד את המרחק המינימלי בין שני קודקודים שיש ביניהם מסלול.
 - (6) נתון גרף לא מכוון. בעץ פורש DFS של הגרף שהורץ מקודקוד התחלה s , יש קודקוד v ולו x בנים. לכן בכל עץ פורש DFS של הגרף שהורץ מקודקוד התחלה v , בהכרח יש לקודקוד v לפחות x בנים.

ב. נתון גרף לא מכוון G ובו 5 קודקודים: a, b, c, d, s .

לאחר הרצת DFS מקודקוד התחלה s התקבל העץ שבסרטוט הנתון:

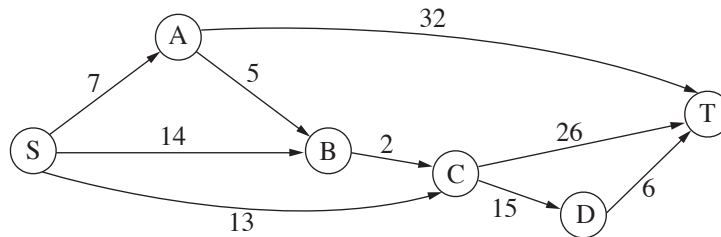


בנוגע לכל אחד משלושת העצים שלפניכם, קבעו אם ייתכן שהוא התקבל מהרצת DFS על אותו הגרף G , מקודקוד התחלה a . נמקו את קביעותיכם.



5. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענו על שני הסעיפים.

א. לפניכם גרף ממושקל G_1 :



(1) איזה אלגוריתם מחזיר את המסלול הקצר (הקל) בין שני קודקודים בגרף ממושקל (אי-שלילי) כלשהו?

מהי סיבוכיות זמן הריצה של האלגוריתם? נמקו את תשובתכם.

נוסף על כך, הריצו את האלגוריתם שכתבתם, בעבור הגרף הנתון G_1 , מצומת S לצומת T. בצעו מעקב שלב אחר שלב ורשמו את המסלול ואת אורכו (סכום משקליו).

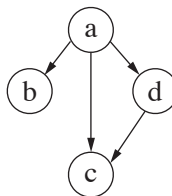
(2) עבור גרף G ממושקל כלשהו, הרצנו את האלגוריתם המוצא את המסלול הקצר (הקל) בין שני קודקודים, וקיבלנו

מסלול העובר דרך הקשתות $E(e_i, e_j, e_m, \dots)$.

– האם המסלול הקצר בין שני הקודקודים יישאר בהכרח זהה אם משקל כל אחת מן הקשתות בגרף יוכפל ב- x ($1 < x$)? נמקו את תשובתכם.

– האם המסלול הקצר בין שני הקודקודים יישאר בהכרח זהה אם משקל כל אחת מן הקשתות בגרף יגדל ב- x ($1 < x$)? נמקו את תשובתכם.

ב. לפניכם גרף מכוון G:



(1) הציגו שתי סריקות DFS מקודקוד התחלה a כך שיתקבלו שני עצים שהגבהים שלהם שונים זה מזה.

(2) הוסיפו מספר מינימלי של קשתות כך ש**שלוש** סריקות DFS מקודקוד a, יתקבלו שלושה עצים שהגבהים שלהם

שונים זה מזה. סרטטו את הגרף לאחר ההוספה, והציגו את שלוש הסריקות.

6. גרף לא מכוון נקרא "גרף-כמעט-עץ" אם קיימת בו קשת שכאשר מורידים אותה מקבלים עץ.

א. סרטטו גרף שיש בו 6 קודקודים והוא "גרף-כמעט-עץ".

ב. נתון "גרף-כמעט-עץ" ובו n קודקודים.

(1) האם אפשר לדעת את מספר הקשתות בגרף זה? נמקו את תשובתכם. אם עניתם שאפשר, ציינו את מספר הקשתות.

(2) האם אפשר לדעת את מספר רכיבי הקשירות בגרף זה? נמקו את תשובתכם. אם עניתם שאפשר, ציינו את מספר

רכיבי הקשירות.

ג. נתון גרף לא מכוון G ובו n קודקודים, המיוצג על ידי רשימת סמיכויות.

(1) כתבו אלגוריתם יעיל המחזיר "אמת" אם הגרף הוא "גרף-כמעט-עץ" ואחרת הוא מחזיר "שקר".

(2) מהי סיבוכיות זמן הריצה של האלגוריתם שכתבתם? נמקו את תשובתכם.

מודלים חישוביים

7. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענו על שני הסעיפים.

א. לפניכם שלוש שפות מעל הא"ב $\{a, b\}$:

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

$$L_2 = \{ a^n b^m a^k \mid n, m, k > 0, n \% 2 = k \% 2 \}$$

$$L_3 = \{ a^n b^m a^k \mid n, m, k > 0, n+m+k < 100 \}$$

לפניכם ארבעה סעיפים (1)–(4). ענו על כולם (בנימוק מספיק הסבר מילולי, אין צורך באוטומט).

(1) בנוגע לכל אחת מן השפות $L_3 - L_1$, כתבו אם היא רגולרית או לא רגולרית. נמקו את תשובתכם.

(2) האם השפה $L_1 \cap L_3$ רגולרית? נמקו את תשובתכם.

(3) האם השפה $L_1 \cap \overline{L_3}$ רגולרית? נמקו את תשובתכם.

(4) האם השפה $L_2 \cap \overline{L_3}$ רגולרית? נמקו את תשובתכם.

ב. נתונה השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{ a^m b^k c^x \mid m, k, x > 0 \text{ וגם } (m \% 2 = k \% 2 \text{ או } m \% 2 = x \% 2) \}$$

הסבר: השפה L מכילה מילים שבהן כל אחת מן האותיות abc מופיעה פעם אחת לפחות, ולפי סדר זה (המופעים של a תחילה ואחר כך של b ואחר כך של c). כמו כן, אם מספר המופעים של האות a הוא זוגי/אי-זוגי, מספר המופעים של האות b אן של האות c יהיה גם הוא זוגי/אי-זוגי בהתאמה.

דוגמה למילה בשפה L היא $abbbccccc$, כי כל אחת מן האותיות abc מופיעה בה, ולפי הסדר הנדרש. כמו כן מספר המופעים של האות a הוא אי-זוגי, וכך גם מספר המופעים של האות c .

דוגמה נוספת למילה בשפה L היא $aaaabbc$, כי כל אחת מן האותיות abc מופיעה בה, ולפי הסדר הנדרש. כמו כן מספר המופעים של האות a הוא זוגי, וכך גם מספר המופעים של האות b .

דוגמה למילה שאינה בשפה L היא $aaabc$. אומנם כל אחת מן האותיות abc מופיעה בה, ולפי הסדר הנדרש, אך מספר המופעים של האות a הוא זוגי, ואילו מספר המופעים של האות b ושל האות c הוא אי-זוגי.

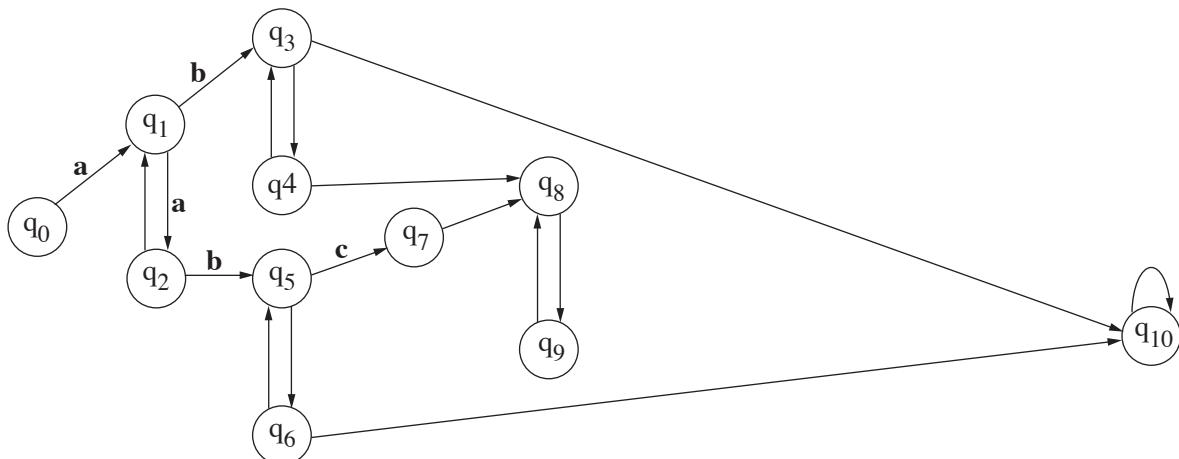
(1) לפניכם חמש מילים: $abbbac$, abc , bbc , $abcc$, $abbcc$.

העתיקו כל אחת מן המילים למחברתכם, וקבעו אם היא שייכת לשפה L או לא שייכת לשפה L . נמקו את קביעותיכם.

(2) נתון לפניכם אוטומט סופי דטרמיניסטי שאינו מלא המקבל את השפה L . באוטומט קיימים כל המצבים וכל המעברים הנדרשים, אך בכמה מן המעברים חסרים סימני הקלט (התווים במעברים), והמצבים המקבלים באוטומט אינם מסומנים כלל.

העתיקו את האוטומט למחברתכם, הוסיפו את סימני הקלט החסרים, וסמנו את המצבים המקבלים.

הערה: האוטומט צריך להישאר סופי דטרמיניסטי שאינו מלא. אין להוסיף בו מצבים או מעברים, ואין לשנות את סימני הקלט המופיעים בו. תשובה שיהיה בה שינוי באוטומט הנתון לא תזוכה בנקודות.



8. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענו על שני הסעיפים.

א. נתונות שלוש שפות L_1, L_2, L_3 מעל הא"ב $\{a, b, c\}$.

לפניכם שלוש טענות. בנוגע לכל אחת מן הטענות, קבעו אם היא נכונה או לא נכונה.

אם היא נכונה – הסבירו מדוע, ואם היא לא נכונה – הפריכו אותה באמצעות דוגמה נגדית: בחרו שפות ספציפיות והראו שעבורן הטענה לא מתקיימת.

$$(1) \quad R(R(L_1) \cdot R(L_2)) = L_1 \cdot L_2$$

$$(2) \quad L_1 \cdot (L_2 \cup L_3) = (L_1 \cdot L_2) \cup (L_1 \cdot L_3)$$

$$(3) \quad L_1 \cap (L_2 \cdot L_3) = (L_1 \cap L_2) \cdot (L_1 \cap L_3)$$

ב. נתונה השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{a^i b^j c^{i-j} \mid i \geq 4, (i-j) \geq 3\}$$

בנו אוטומט מחסנית דטרמיניסטי המקבל את השפה L .

9. כתבו מכונת טיורינג המקבלת בתחילת הסרט מספר אונרי הגדול מאחד, ומחזירה מספר חדש כמפורט לפניכם:

- אם המספר שהתקבל זוגי – המכונה מחזירה את המספר המקורי חלקי 2.
- אם המספר שהתקבל אי-זוגי – המכונה מחזירה את המספר המקורי פחות 1.

הנחיות:

– אין צורך לשמור על הקלט (אפשר לשנות את הקלט לסימנים שונים).

– המספר המוחזר יופיע במקום כלשהו בסרט בין שני סימני \$.

– אין צורך לבדוק את תקינות הקלט.

דוגמה למספר זוגי

הסרט לפני ההרצה:

–	1	1	1	1	1	1	△	△	△	△	△
---	---	---	---	---	---	---	---	---	---	---	---

הסרט לאחר ההרצה:

...	\$	1	1	1	\$
-----	----	---	---	---	----

דוגמה למספר אי-זוגי

הסרט לפני ההרצה:

–	1	1	1	1	1	△	△	△	△	△
---	---	---	---	---	---	---	---	---	---	---

הסרט לאחר ההרצה:

...	\$	1	1	1	1	\$
-----	----	---	---	---	---	----

שימו לב: השאלות 10–12 הן גם בעבור הפותרים בשפת Java וגם בעבור הפותרים בשפת C#. לכל שאלה נוסח בשפת Java ונוסח בשפת C#.

10. לשאלה זו שני נוסחים: בשפת Java בעמודים 14–15 ובשפת C# בעמודים 16–17.

לפותרים בשפת Java

בגן החיות "חיים בשלום" פותחה מערכת ממוחשבת ובה שומרים נתונים על בעלי החיים שבגן. המערכת כוללת את המחלקות האלה: בעל חיים – **Animal**, ציפור – **Bird**, תוכי – **Parrot**, דג – **Fish**, נחש – **Snake**. נתון חלק ממבנה המחלקות במערכת הממוחשבת:

```
public class Animal {
    private String color;
    private int weight;
    public Animal (String color, int weight)
    {
        this.color = color;
        this.weight = weight;
    }
    public String toString() {
        return "My color is " + this.color + "! And I weigh " + weight + " kilos!";
    }
}

public class Bird {
    private String color;
    private int weight;
    private boolean isFlying;
    public void zoo () {
        System.out.println ("Hello");
    }
}

public class Parrot {
    private String color;
    private int weight;
    private boolean isFlying;
    private boolean isTalking;
    public void zoo () {
        System.out.println ("Hello");
    }
}

public class Fish {
    private String color;
    private int weight;
    private String waterType;    // דגים/דג מים מתוקים
}

public class Snake {
    private String color;
    private int weight;
    private int length;
    private boolean isVenomous;    // האם הנחש ארסי?
}
```

הניחו שיש פעולות get ו-set לתכונות המחלקות.

(שימו לב: המשך השאלה בעמוד הבא.)

א. כתבו מחדש את הכותרות, התכונות והפעולות של המחלקות: **Fish**, **Parrot**, **Bird** ו- **Snake**, לפי עקרונות תכנות מונחה עצמים (אין לשנות את המחלקה Animal). בסעיף זה אין צורך לכתוב פעולות בונות. הניחו שהפעולות get ו- set קיימות.

נוסף על כך, סרטטו תרשים הייררכייה בין כל המחלקות. יש לסמן ירושה באמצעות החץ \triangleleft .

ב. נתונה המחלקה Tester:

```
public class Tester {
    public static void main (String[] args) {
        Animal[] animals = new Animal [5];
        animals [0] = new Bird ("white", 4, false);
        animals [1] = new Fish ("blue", 3, "sweet water");
        animals [2] = new Parrot ("brown", 12, true, true);
        animals [3] = new Snake ("gray", 2, 6, true);
        animals [4] = new Snake ("black", 3, 4, false);
        /***/
    }
}
```

הוסיפו פעולות בונות למחלקות **Fish**, **Parrot**, **Bird** ו- **Snake**, המאתחלות את תכונות המחלקות (כך שהפעולה main תרוץ ללא שגיאות).

הערה: אין לשנות את הפעולה הבונה הנתונה של **Animal**.

ג. (1) בפעולה main נוסף קטע הקוד שלפניכם (במקום המסומן ב- ***):

```
for (int i = 0; i < animals.length; i++) {
    System.out.println (animals[i]); }
```

קטע הקוד תקין ומדפיס את הפלט שלפניכם:

```
I'm a bird! My color is white! And I weigh 4 kilos!
My color is blue! And I weigh 3 kilos!
I'm a parrot! My color is brown!
I'm a snake! I'm venomous, be careful! My color is gray! And I weigh 2 kilos!
I'm a snake! I'm not venomous! My color is black! And I weigh 3 kilos!
```

הוסיפו את הפעולה toString במחלקות (רק היכן שיש צורך) לפי עקרונות תכנות מונחה עצמים, כך שהפלט יהיה מתאים.

הערה: אין לשנות את הפעולה toString במחלקה **Animal**. תשובה שבה יהיה שינוי לא תזוכה בנקודות.

(2) במחלקה Tester נוספה הפעולה שלפניכם:

```
public static void hello (Animal [] arr) {
    for (int i = 0; i < arr.length; i++) {
        arr[i].zoo(); }
}
```

אם הפעולה תקינה, כתבו את הפלט שלה עבור המערך animals (הנתון לעיל), ואם הפעולה לא תקינה – תקנו אותה (כך שהקריאה ל- zoo בתוך הלולאה תהיה תקינה) וכתבו את הפלט עבור המערך animals. הערות: – אין לשנות את המחלקות עצמן. תשובה שבה יהיה שינוי במחלקות לא תזוכה בנקודות.

– הפעולה צריכה לרוץ ללא שגיאות עבור כל מערך מטיפוס Animal.

לפותרים בשפת C#

בגן החיות "חיים בשלום" פותחה מערכת ממוחשבת ובה שומרים נתונים על בעלי החיים שבגן. המערכת כוללת את המחלקות האלה: בעל חיים – **Animal**, ציפור – **Bird**, תוכי – **Parrot**, דג – **Fish**, נחש – **Snake**. נתון חלק ממבנה המחלקות במערכת הממוחשבת:

```
public class Animal {
    private string color;
    private int weight;
    public Animal (string color, int weight)
    {
        this.color = color;
        this.weight = weight;
    }
    public override string ToString() {
        return "My color is " + this.color + "! And I weigh " + weight + " kilos!";
    }
}

public class Bird {
    private string color;
    private int weight;
    private bool isFlying;
    public void Zoo () {
        Console.WriteLine ("Hello");
    }
}

public class Parrot {
    private string color;
    private int weight;
    private bool isFlying;
    private bool isTalking;
    public void Zoo () {
        Console.WriteLine ("Hello");
    }
}

public class Fish {
    private string color;
    private int weight;
    private string waterType;    // דג ים/דג מים מתוקים
}

public class Snake {
    private string color;
    private int weight;
    private int length;
    private bool isVenomous;    // האם הנחש ארסי?
}
```

הניחו שיש פעולות Get ו- Set לתכונות המחלקות.

(שימו לב: המשך השאלה בעמוד הבא.)

א. כתבו מחדש את הכותרות, התכונות והפעולות של המחלקות: **Fish**, **Parrot**, **Bird** ו- **Snake**, לפי עקרונות תכנות מונחה עצמים (אין לשנות את המחלקה **Animal**). בסעיף זה אין צורך לכתוב פעולות בונות. הניחו שהפעולות **Get** ו- **Set** קיימות.

נוסף על כך, סרטטו תרשים הייררכייה בין כל המחלקות. יש לסמן ירושה באמצעות החץ \triangleleft .

ב. נתונה המחלקה **Tester**:

```
public class Tester {
    public static void Main (string[] args) {
        Animal[] animals = new Animal [5];
        animals [0] = new Bird ("white", 4, false);
        animals [1] = new Fish ("blue", 3, "sweet water");
        animals [2] = new Parrot ("brown", 12, true, true);
        animals [3] = new Snake ("gray", 2, 6, true);
        animals [4] = new Snake ("black", 3, 4, false);
        /**
    }
```

הוסיפו פעולות בונות למחלקות **Fish**, **Parrot**, **Bird** ו- **Snake**, המאתחלות את תכונות המחלקות (כך שהפעולה **Main** תרוץ ללא שגיאות).

הערה: אין לשנות את הפעולה הבונה הנתונה של **Animal**.

ג. (1) בפעולה **Main** נוסף קטע הקוד שלפניכם (במקום המסומן ב- *******):

```
for (int i = 0; i < animals.Length; i++) {
    Console.WriteLine (animals[i]); }
```

קטע הקוד תקין ומדפיס את הפלט שלפניכם:

```
I'm a bird! My color is white! And I weigh 4 kilos!
My color is blue! And I weigh 3 kilos!
I'm a parrot! My color is brown!
I'm a snake! I'm venomous, be careful! My color is gray! And I weigh 2 kilos!
I'm a snake! I'm not venomous! My color is black! And I weigh 3 kilos!
```

הוסיפו את הפעולה **ToString** במחלקות (רק היכן שיש צורך) לפי עקרונות תכנות מונחה עצמים, כך שהפלט יהיה מתאים.

הערה: אין לשנות את הפעולה **ToString** במחלקה **Animal**. תשובה שבה יהיה שינוי לא תזוכה בנקודות.

(2) במחלקה **Tester** נוספה הפעולה שלפניכם:

```
public static void Hello (Animal [] arr) {
    for (int i = 0; i < arr.Length; i++) {
        arr[i].Zoo(); }
}
```

אם הפעולה תקינה, כתבו את הפלט שלה עבור המערך **animals** (הנתון לעיל), ואם הפעולה לא תקינה – תקנו אותה (כך שהקריאה ל- **Zoo** בתוך הלולאה תהיה תקינה) וכתבו את הפלט עבור המערך **animals**. הערה: – אין לשנות את המחלקות עצמן. תשובה שבה יהיה שינוי במחלקות לא תזוכה בנקודות.

– הפעולה צריכה לרוץ ללא שגיאות עבור כל מערך מטיפוס **Animal**. המשך בעמוד 18/

11. לשאלה זו שני נוסחים: בשפת **Java** בעמודים 18–19 ובשפת **C#** בעמודים 20–21.

לפותרים בשפת Java

נתונות המחלקות One, Two, Three :

```
public class One {
    public static int count = 0;
    private int number;
    public One() {
        count++;
        number = count;
    }
    public One (int num) {
        number = num;
    }
    public String toString() {
        return count + ", " + number;
    }
}

-----

public class Two extends One {
    private String strTwo;
    public Two() {
        strTwo = "Fast";
    }
    public Two (String s) {
        super (15);
        strTwo = s;
    }
    public String toString() {
        return super.toString() + " " + strTwo;
    }
}

-----

public class Three extends Two {
    private String strThree;
    public Three (String s) {
        strThree = s;
    }
    public String toString() {
        return super.toString() + " " + strThree;
    }
}
```

(שימו לב: המשך השאלה בעמוד הבא.)

א. נתונה המחלקה Test :

```
public class Test
{
    public static void main (String[] args)
    {
        One[] arr=new One[6];
        arr[0]= ... // (1)
        arr[1]= ... // (2)
        arr[2]= ... // (3)
        arr[3]= ... // (4)
        arr[4]= ... // (5)
        arr[5]= ... // (6)
        for (int i = 0; i < arr.length; i++) {
            System.out.println (arr[i]);
        }
    }
}
```

השלימו את השורות (1)–(6) כך שהפלט יהיה:

```
4, 1 Fast
4, 2 Fast Car
4, 3 Fast Horse
4, 15 Small
4, 15 Tall
4, 4
```

ב. הכותרת של המחלקה Three השתנתה לכוותרת:

public class Three extends One

האם שינוי זה יגרום לשגיאה? אם כן – הסבירו את השגיאה, אם לא – רשמו מה יהיה הפלט של הפעולה main אחרי השינוי (השורות החסרות הן כפי שכתבתם בסעיף א).
הערה: אין שינויים נוספים במחלקה.

לפותרים בשפת C#

נתונות המחלקות One, Two, Three :

```
public class One {
    public static int count = 0;
    private int number;
    public One() {
        count++;
        number = count;
    }
    public One (int num) {
        number = num;
    }
    public override string ToString() {
        return count + ", " + number;
    }
}

-----

public class Two : One {
    private string strTwo;
    public Two() {
        strTwo = "Fast";
    }
    public Two (string s) : base (15) {
        strTwo = s;
    }
    public override string ToString() {
        return base.ToString() + " " + strTwo;
    }
}

-----

public class Three : Two {
    private string strThree;
    public Three (string s) {
        strThree = s;
    }
    public override String ToString() {
        return base.ToString() + " " + strThree;
    }
}
```

(שימו לב: המשך השאלה בעמוד הבא.)

א. נתונה המחלקה Test :

```

public class Test
{
    public static void Main (string[] args)
    {
        One[] arr=new One[6];
        arr[0]= ... // (1)
        arr[1]= ... // (2)
        arr[2]= ... // (3)
        arr[3]= ... // (4)
        arr[4]= ... // (5)
        arr[5]= ... // (6)
        for (int i = 0; i < arr.Length; i++) {
            Console.WriteLine (arr[i]);
        }
    }
}

```

השלימו את השורות (1)–(6) כך שהפלט יהיה:

```

4, 1 Fast
4, 2 Fast Car
4, 3 Fast Horse
4, 15 Small
4, 15 Tall
4, 4

```

ב. הכותרת של המחלקה Three השתנתה לכותרת:

```

public class Three : One

```

האם שינוי זה יגרום לשגיאה? אם כן – הסבירו את השגיאה, אם לא – רשמו מה יהיה הפלט של הפעולה Main אחרי השינוי (השורות החסרות הן כפי שכתבתם בסעיף א).
הערה: אין שינויים נוספים במחלקה.

12. לשאלה זו שני נוסחים: בשפת **Java** בעמודים 22–23 ובשפת **C#** בעמודים 24–25.

לפותרים בשפת Java

נתונות המחלקות **A, B, C, D**:

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

```
public class B extends A {  
    public B (int i) { super (i + 1); }  
    public boolean foo (Object myObject) {  
        System.out.println ("foo1");  
        return ((myObject instanceof B) && (this.i == ((B)myObject).i));  
    }  
    public boolean foo (B myB, int num) {  
        System.out.println ("foo2");  
        return (this.i + myB.i) < num;  
    }  
}
```

```
public class C extends B {  
    public C (int i) { super (i); }  
    public boolean foo (int num) {  
        System.out.println ("foo3");  
        return (this.i != num);  
    }  
}
```

```
public class D extends B {  
    public D (int i) { super (i * 2); }  
    public boolean foo (B myB, int num) {  
        System.out.println ("foo4");  
        return super.foo (myB, 20);  
    }  
}
```

(שימו לב: המשך השאלה בעמוד הבא.)

א. סרטטו את תרשימים ההייררכייה של המחלקות A, B, C, D. יש לסמן ירושה באמצעות החץ .

ב. נתונה המחלקה Tester:

```
public class Tester {
    public static void main (String[] args) {
        A a = new A (1);
        B b = new B (2);
        C c = new C (3);
        D d = new D (4);
        B bd = new D (5);
        A ac = new C (6);
        /***
    }
```

(1) סרטטו את העצמים שנוצרו בפעולה main.

(2) במקום המסומן ב- *** נוסף קטע הקוד התקין שלפניכם:

```
System.out.println (c.foo(5)); // (1)
System.out.println (d.foo(a)); // (2)
System.out.println (bd.foo(b)); // (3)
System.out.println (bd.foo(b, 1)); // (4)
System.out.println (((C)ac).foo(c)); // (5)
```

כתבו את הפלט של הפעולה (כתבו את מספר השורה המשויך לכל פלט).

C# לפותרים בשפתנתונות המחלקות **A, B, C, D**:

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

```
public class B : A {  
    public B (int i) : base (i + 1) { }  
    public bool Foo (Object myObject) {  
        Console.WriteLine ("Foo1");  
        return ((myObject is B) && (this.i == ((B)myObject).i));  
    }  
    public virtual bool Foo (B myB, int num) {  
        Console.WriteLine ("Foo2");  
        return (this.i + myB.i) < num;  
    }  
}
```

```
public class C : B {  
    public C (int i) : base (i) { }  
    public bool Foo (int num) {  
        Console.WriteLine ("Foo3");  
        return (this.i != num);  
    }  
}
```

```
public class D : B {  
    public D (int i) : base (i * 2) { }  
    public override bool Foo (B myB, int num) {  
        Console.WriteLine ("Foo4");  
        return base.Foo (myB, 20 );  
    }  
}
```

(שימו לב: המשך השאלה בעמוד הבא.)

א. סרטטו את תרשימים ההייררכייה של המחלקות A, B, C, D. יש לסמן ירושה באמצעות החץ .

ב. נתונה המחלקה Tester :

```
public class Tester {
    public static void Main (string[] args) {
        A a = new A (1);
        B b = new B (2);
        C c = new C (3);
        D d = new D (4);
        B bd = new D (5);
        A ac = new C (6);
        /***
    }
}
```

(1) סרטטו את העצמים שנוצרו בפעולה Main.

(2) במקום המסומן ב- *** נוסף קטע הקוד התקין שלפניכם:

```
Console.WriteLine (c.Foo(5)); // (1)
Console.WriteLine (d.Foo(a)); // (2)
Console.WriteLine (bd.Foo(b)); // (3)
Console.WriteLine (bd.Foo(b, 1)); // (4)
Console.WriteLine (((C)ac).Foo(c)); // (5)
```

כתבו את הפלט של הפעולה (כתבו את מספר השורה המשויך לכל פלט).

בהצלחה!