

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 3
מספר השאלון: 899271
שם התלמיד/ה: _____

מדעי המחשב

מבני נתונים ומסלול - 4-5 יחידות לימוד

הוראות

א. משך הבחינה: שלוש שעות.

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

פרק ראשון - 2×25 - 50 נקודות

פרק שני - 2×25 - 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.

השאלות

פרק ראשון

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

1. (25 נקודות)

שימו לב: בשאלה זו אפשר להשתמש בפעולות של המחלקות תור, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

בחנות נרשם קוד המוצר של כל מכירה במהלך היום, לפי סדר המכירות, בתור q של מספרים שלמים. מוצר שנמכר כמה פעמים מופיע בתור כמה פעמים.

"מוצר בודד" הוא מוצר שהקוד שלו מופיע בתור **פעם אחת בדיק** (המוצר נמכר פעם אחת בלבד).

ממשו את הפעולה החיצונית שלפניכם:

Java - `public static int countSingles (Queue<Integer> q)`

#C - `public static int CountSingles (Queue<int> q)`

הפעולה מקבלת את התור q של קודי המכירות, ומחזירה את מספר ה**מוצרים הבודדים** בתור.

דוגמה: עבור התור q שלפניכם הפעולה תחזיר 3, כי המוצרים הבודדים הם 5, 9, 4 (המוצר 7 נמכר שלוש פעמים, והמוצר 2 נמכר פעמיים).

	ראש התור								
q		7	2	7	5	9	2	7	4

הערות:

- הניחו שבתור q יש שני איברים לפחות.

- אין צורך לשמור על התור q .

- אין להשתמש בשאלה זו במערך וברשימה מקושרת.

ממשו את הפעולה `countSingles/CountSingles` המתוארת לעיל.

2. (25 נקודות)

נוסח Java

שימו לב: לשאלה זו שני נוסחים: בשפת Java ובשפת #C.

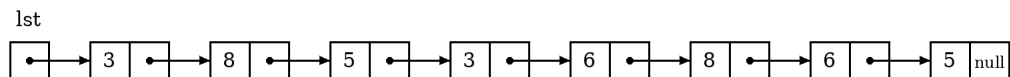
לפותרים בשפת Java

א. (1) נתונה הפעולה `what`:

```
public static int what (Node<Integer> lst, int x)
{
    if (lst == null)
        return 0;
    if (lst.getValue() == x)
        return 1;
```

```
int r = what (lst.getNext(), x);
if (r == 0)
    return 0;
return r + 1;
}
```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה `what (lst, 6)`, ורשמו את הערך שהפעולה מחזירה.

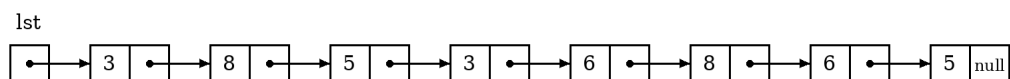
(2) מה עושה הפעולה `what`? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה `what`? נמקו את תשובתכם.

ב. (1) נתונה הפעולה `guess`:

```
Java
public static int guess (Node<Integer> lst)
{
    if (lst == null)
        return 0;
    int d = what (lst.getNext(), lst.getValue());
    int r = guess (lst.getNext());
    if (d != 0 && (r == 0 || d < r))
        return d;
    return r;
}
```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה `guess (lst)`, ורשמו את הערך שהיא מחזירה.

בסעיף זה אין צורך לעקוב אחר הפעולה `what`.

(2) מה עושה הפעולה `guess`? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה `guess`? נמקו את תשובתכם.

נוסח C#

שימו לב: לשאלה זו שני נוסחים: בשפת Java ובשפת C#.

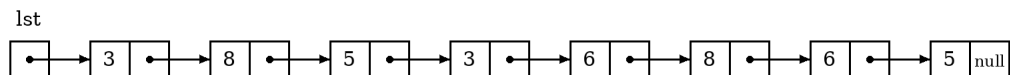
לפותרים בשפת C#

א. (1) נתונה הפעולה `What`:

```
C#
public static int What (Node<int> lst, int x)
{
    if (lst == null)
        return 0;
    if (lst.GetValue() == x)
        return 1;
    int r = What (lst.GetNext(), x);
    if (r == 0)
        return 0;
}
```

```
return r + 1;
}
```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה What (lst, 6), ורשמו את הערך שהפעולה מחזירה.

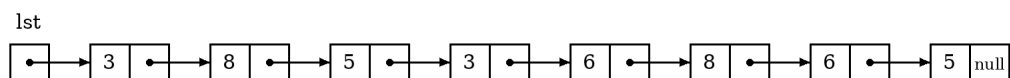
(2) מה עושה הפעולה What? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה What? נמקו את תשובתכם.

ב. (1) נתונה הפעולה Guess:

```
C#
public static int Guess (Node<int> lst)
{
    if (lst == null)
        return 0;
    int d = What (lst.GetNext(), lst.GetValue());
    int r = Guess (lst.GetNext());
    if (d != 0 && (r == 0 || d < r))
        return d;
    return r;
}
```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה Guess (lst), ורשמו את הערך שהיא מחזירה.

בסעיף זה אין צורך לעקוב אחר הפעולה What.

(2) מה עושה הפעולה Guess? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה Guess? נמקו את תשובתכם.

3. (25 נקודות)

ב. כתבו פעולה חיצונית ששמה isZigzagK בשפת Java או IsZigzagK בשפת C#, המקבלת עץ בינרי - root מטיפוס שלם שאינו ריק, וערך K שאינו שלילי, ומחזירה true אם העץ הוא "עץ K-זיגזג", ואם לא - היא מחזירה false .
אפשר להיעזר בפעולת עזר.

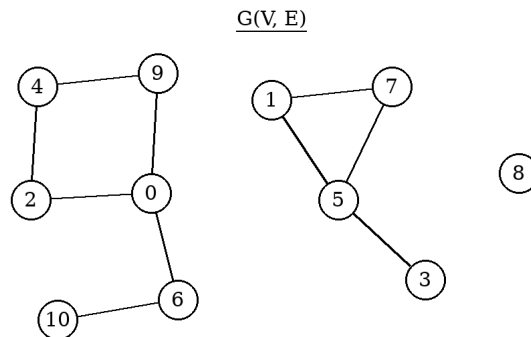
יש לענות על **שתי** שאלות במסלול שלמדתם (לכל שאלה - 25 נקודות).

נכתב ונבדק על ידי צוותי GOOL בגריות ו-GuyAI · www.guyai.co.il

א. כתבו אלגוריתם שמקבל קודקוד בגרף v_j , ומספר שלם $k \geq 1$, ומחזיר את כל הקודקודים (חוץ מ- v_j עצמו) שהמרחק שלהם מ- v_j הוא **לכל היותר** k , כלומר את כל הקודקודים שיש מסלול בינם ובין v_j שיש בו k קשתות או פחות.

הערה: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

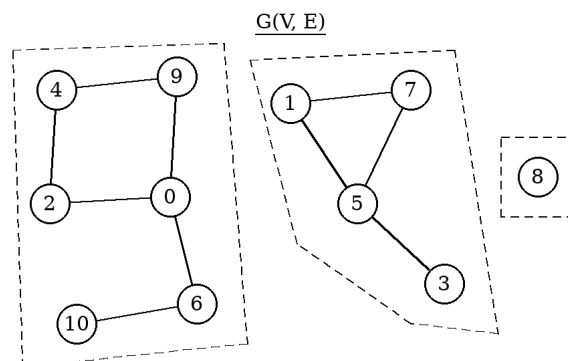
דוגמה: עבור הגרף שלפניכם, הקודקוד 4 ו- $k = 2$, האלגוריתם יחזיר את הקודקודים 0, 2, 9.



הסבר: המרחק מקודקוד 4 אל הקודקודים 2 ו-9 הוא קשת אחת, והמרחק אל קודקוד 0 הוא 2 קשתות. המרחק אל קודקוד 6 הוא 3 קשתות, ולכן הוא אינו מוחזר, ואל שאר הקודקודים אין מסלול מקודקוד 4.

ב. "רכיב קשירות" בגרף לא מכוון $G(V, E)$ הוא קבוצת קודקודים שבה בין כל שני קודקודים יש מסלול, ואין שום קשת היוצאת מקודקוד בקבוצה לקודקוד שאינו בקבוצה. קודקוד שאין קשת בינו ובין שום קודקוד אחר יהיה בקבוצה משלו.

דוגמה: בגרף שבדוגמה לעיל יש שלושה רכיבי קשירות, והם מסומנים בקו מקווקו.



מנהל הרשת רוצה להוסיף כבלים (קשתות) כך שבין כל שני מחשבים יהיה מסלול, כלומר שהגרף יהיה קשיר.

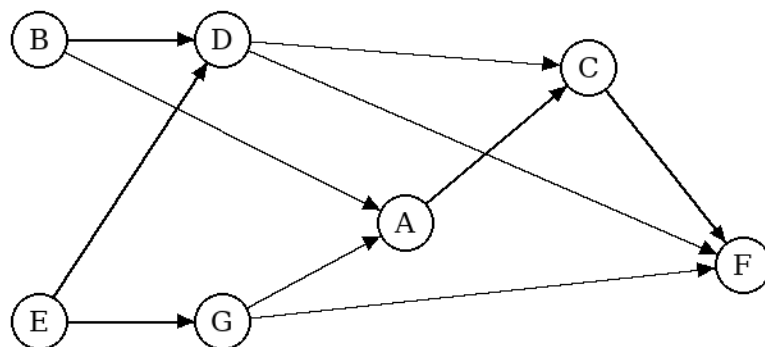
כתבו אלגוריתם המחזיר את המספר **המינימלי** של קשתות שיש להוסיף לגרף $G(V, E)$ כדי שיהיה קשיר, ורשימה של קשתות כאלה שאפשר להוסיף.

למשל עבור הדוגמה שלעיל, האלגוריתם יחזיר את המספר 2 ואת הקשתות (1, 0) ו- (8, 0) .

הערה: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

5. (25 נקודות)

בפרויקט יש שבע משימות: A, B, C, D, E, F, G. יש משימות שאפשר להתחיל רק אחרי שמשימות אחרות הסתיימו. את הדרישות מציג הגרף המכוון $G = (V, E)$ שלפניכם: קשת $X \rightarrow Y$ פירושה שאת המשימה Y אפשר להתחיל רק אחרי שהמשימה X הסתיימה.



$V = \{A, B, C, D, E, F, G\}$	צמתי הגרף
$E = \{(A \rightarrow C), (B \rightarrow A), (B \rightarrow D), (C \rightarrow F), (D \rightarrow C), (D \rightarrow F), (E \rightarrow D), (E \rightarrow G), (G \rightarrow A), (G \rightarrow F)\}$	קשתות הגרף

- א. הצומת M מוגדר "שכן יוצא" של הצומת U, אם יש בגרף קשת $U \rightarrow M$.
 לפניכם טבלה המציגה את השכנים היוצאים של כל צומת בגרף G, מסודרים לפי סדר האלף-בית.
 השכנים היוצאים של B ושל E נתונים בטבלה.
 השלימו את הטבלה. (אם לצומת אין שכנים יוצאים, כתבו -).

G	F	E	D	C	B	A	U
		D, G			A, D		M

- ב. נתון האלגוריתם **Order** (G), שמחשב סדר לביצוע המשימות.

שלב אתחול נתונים באלגוריתם:

1. נגדיר מערך indeg בגודל 7, ובו נשמור בעבור כל צומת את מספר הקשתות שנכנסות אליו (דרגת הכניסה שלו). בהתחלה:

G	F	E	D	C	B	A	:indeg
1	3	0	2	2	0	2	

2. נגדיר מערך level בגודל 7 (בהתחלה בכל התאים במערך מופיע הערך 0).

G	F	E	D	C	B	A	:level
0	0	0	0	0	0	0	

3. נגדיר תור q , שאליו נכניס, לפי סדר האלף-בית, את כל הצמתים שבתא שלהם במערך indeg מופיע הערך 0 (כלומר את B ואחריו את E).

4. נגדיר רשימה R , שבהתחלה היא ריקה.

שלב הרצת האלגוריתם:

1. כל עוד התור q אינו ריק:

• נוציא את הצומת הנמצא בראש התור, ונקרא לו U. נוסיף את U לסוף הרשימה R.

. בעבור כל שכן יוצא - M של הצומת U, לפי הסדר שבטבלה מסעיף א:

- נקטין ב-1 את הערך בתא של M במערך $indeg$.
 - אם $level[U] + 1$ גדול מן הערך $level[M]$, נכתוב בתא של M במערך $level$ את הערך $level[U] + 1$.
 - אם הערך בתא של M במערך $indeg$ הוא עכשיו 0, נכניס את M לתור.
2. כשהתור ריק: אם ברשימה R יש 7 צמתים, נחזיר את R; אחרת נחזיר null. האלגוריתם

מסתיים.

עקבו בטבלת מעקב אחרי ריצת האלגוריתם Order על הגרף הנתון G. המעקב צריך לכלול בכל איטרציה את הפריטים שלהלן:

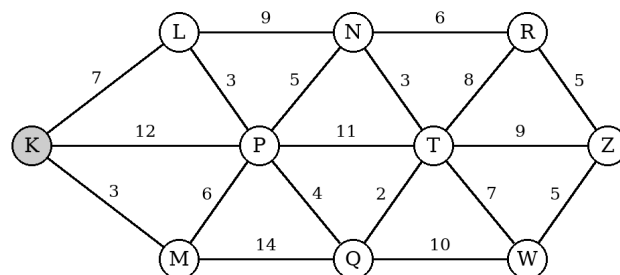
- מצב התור
- הצומת U שהוצאנו מן התור
- מערך $indeg$
- מערך $level$
- הרשימה R

התור	U	$indeg$	A	B	C	D	E	F	G	A	level	B	C	D	E	F	G	R
←[]																		
←[]																		

ג. כתבו את הרשימה שהאלגוריתם מחזיר.

6. (25 נקודות)

לפניכם הגרף הממושקל G. הקודקודים הם צמתי דרכים בעיר, והמשקל של כל קשת הוא זמן הנסיעה (בדקות) בין שני הצמתים:



א. (1) כתבו אלגוריתם המקבל גרף ממושקל (אי-שלילי) לא מכוון כלשהו, שבו n קודקודים מ- v_0 עד v_{n-1} , ושני קודקודים בגרף - v_i ו- v_j . האלגוריתם ימצא את המסלולים הקלים ביותר מקודקוד v_i אל שאר הקודקודים שבגרף, ואחר כך ידפיס את המסלול הקל ביותר מ- v_j אל v_i - את הקודקודים שבמסלול לפי הסדר, מ- v_j ועד v_i . אם אין מסלול מ- v_j אל v_i , האלגוריתם ידפיס "אין מסלול".

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

- ב. (1) בעבור גרף G הנתון, מצאו בעזרת האלגוריתם שכתבתם את המסלול הקל ביותר מקודקוד K לכל אחד מן הקודקודים, וסרטטו טבלת מעקב כמפורט:
- המעקב יכלול בכל איטרציה את קבוצת הקודקודים הקבועים (שכבר ביקרנו בהם) - P ואת קבוצת הקודקודים הזמניים (שבהם עדיין לא ביקרנו) - T . נוסף על כך, בעבור כל קודקוד יצוין אורך המסלול עד אליו וזהות הקודקוד הקודם לו ("הורה" שלו).
- מה ידפיס האלגוריתם עבור $v_t = W$?
- (2) בעבור גרף G הנתון, סרטטו את עץ המסלולים הקלים ביותר (מקודקוד K).

פרק שני (50 נקודות) - מודלים חישוביים

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

7. (25 נקודות)

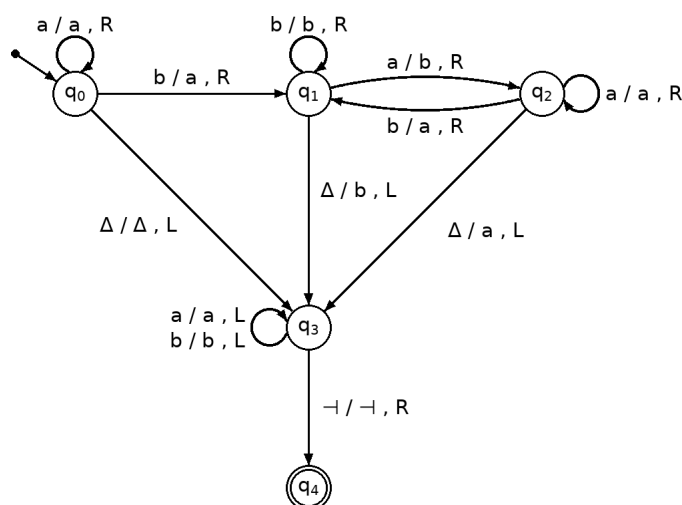
- L היא שפת כל המילים מעל הא"ב $\{a, b, c\}$ המקיימות את כל הדרישות שלפניכם:
- המילה אינה ריקה, והאות הראשונה במילה זהה לאות האחרונה במילה (במילה בת אות אחת, אות זו היא גם הראשונה וגם האחרונה).
 - מספר המופעים של האות c במילה מתחלק ב-3 בלי שארית.
 - המילה אינה מכילה את הרצף ba .
 - אורך המילה הוא מספר אי-זוגי.
- א. (1) הביאו דוגמה אחת למילה השייכת לשפה L , שמופיעות בה כל שלוש האותיות a, b, c .
- (2) הביאו דוגמה אחת למילה שאינה שייכת לשפה L , אף שהיא מקיימת שלוש מן הדרישות. ציינו איזו דרישה אינה מתקיימת בה.
- ב. הוכיחו שהשפה L רגולרית.
- הערה: אפשר להיעזר בתכונות סגירות.

8. (25 נקודות)

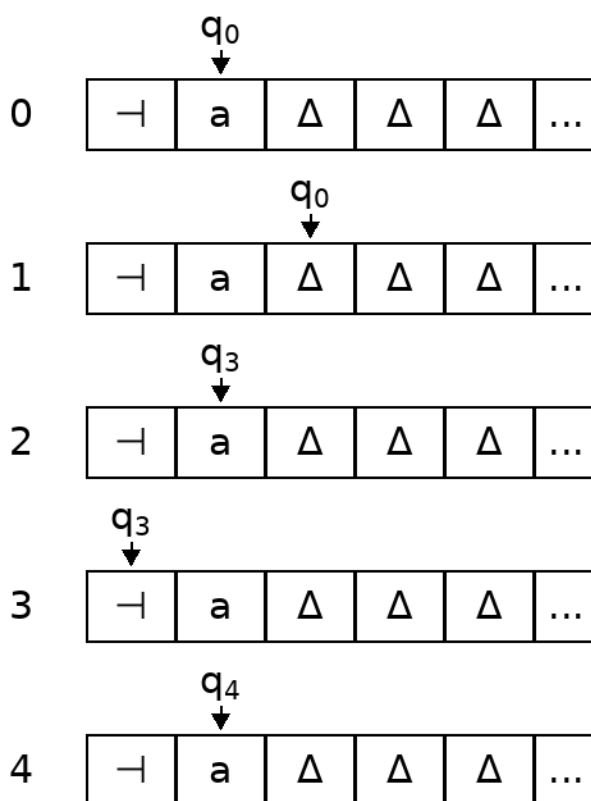
- בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענו על שני הסעיפים.
- א. (1) נתונות שלוש שפות L_1, L_2, L_3 מעל הא"ב $\{a, b\}$. הסימון $R(L)$ מציין את השפה ההפוכה של L , והסימון L^* מציין את סגור קלייני של L (פעולת הכוכב).
- לפניכם שלוש טענות. בנוגע לכל אחת מן הטענות, קבעו אם היא נכונה או לא נכונה.
- אם היא נכונה - הסבירו מדוע, ואם היא לא נכונה - הפריכו אותה באמצעות דוגמה נגדית:
- בחרו שפות ספציפיות והראו שעבורן הטענה לא מתקיימת.
- $$L_1^* \cdot L_2 = (L_1 \cdot L_2)^* \quad (1)$$
- $$R(L_1 \cup L_2) = R(L_1) \cup R(L_2) \quad (2)$$
- $$L_3 = (L_1 \cdot L_3) \cap (L_2 \cdot L_3) \cdot (L_1 \cap L_2) \quad (3)$$
- ב. נתונה השפה L מעל הא"ב $\{a, b, c\}$:
- $$L = \{a^n b^m c^{n+2m} \mid n \geq 2, m \geq 0\}$$
- בנו אוטומט מחסנית דטרמיניסטי המקבל את השפה L .

9. (25 נקודות)

לפניכם מכונת טיורינג המקבלת מילים מעל הא"ב $\{a, b\}$ (מקרא: R - ימין, L - שמאל).



בסעיף א שלפניכם תתבקשו לבצע מעקב אחר פעולת המכונה. במעקב יש להראות בכל שלב את הסרט, מעל איזה תא ממוקם הראש הקורא, ובאיזה מצב q_0 - q_4 נמצאים. להלן דוגמה לאופן שבו יש לבצע מעקב בעבור המכונה הנתונה והמילה a.



- א. (1) עקבו אחרי פעולת המכונה בעבור המילה b. יש להציג את המעקב.
 (2) עקבו אחרי פעולת המכונה בעבור המילה abb. יש להציג את המעקב.
 (3) הציגו את הסרט המתקבל בגמר הפעולה בעבור המילה bbab. אין צורך להציג את המעקב.
- ב. (1) האם קיימות שתי מילים **שוונות** שהמכונה הופכת לאותה מילה (כלומר, בגמר הפעולה על הסרט אותה מילה)? אם כן - כתבו דוגמה. אם לא - נמקו את תשובתכם.

תכנות מונחה עצמים

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

10. (25 נקודות)**נוסח Java**

במוזיאון "בית האוצרות" פותחה מערכת ממוחשבת לניהול תערוכות, ובה המחלקות האלה:
Exhibit - מוצג, **Painting** - ציור, **Sculpture** - פסל, **Photo** - צילום, **Loan** - השאלה של מוצג לתערוכה,

Exhibition - תערוכה.

להלן פירוט תכונות המחלקות:

למחלקה **Exhibit** (מוצג) שתי תכונות: **title** - שם המוצג (מטיפוס מחרוזת), **year** - שנת היצירה של המוצג (מטיפוס שלם).

למחלקה **Painting** (ציור) שלוש תכונות: **title** - שם המוצג (מטיפוס מחרוזת), **year** - שנת היצירה של המוצג (מטיפוס שלם), **technique** - טכניקת הציור, למשל "שמן" או "צבעי מים" (מטיפוס מחרוזת).

למחלקה **Sculpture** (פסל) שלוש תכונות: **title** - שם המוצג (מטיפוס מחרוזת), **year** - שנת היצירה של המוצג (מטיפוס שלם), **weight** - משקל הפסל בק"ג (מטיפוס שלם).

למחלקה **Photo** (צילום) שלוש תכונות: **title** - שם המוצג (מטיפוס מחרוזת), **year** - שנת היצירה של המוצג (מטיפוס שלם), **isColor** - האם הצילום צבעוני (מטיפוס בוליאני). אם הצילום צבעוני - **true**, ואם הוא בשחור-לבן - **false**.

למחלקה **Loan** (השאלה) שתי תכונות: **lender** - שם המשאיל (מטיפוס מחרוזת), **exhibit** - המוצג המושאל (מטיפוס **Exhibit**). בכל השאלה מושאל מוצג אחד בלבד.

למחלקה **Exhibition** (תערוכה) שתי תכונות: **arrL** - מערך של השאלות (מטיפוס **Loan**), **count** - כמות ההשאלות השמורות במערך (מטיפוס שלם).

ההשאלות מופיעות ברצף מתחילת המערך ללא **null** ביניהן, אך ייתכן שהמערך אינו מלא עד הסוף. לכל המוצגים בתערוכה יש שמות שונים.

א. (1) סרטטו תרשים הייררכייה בין כל המחלקות.

יש לסמן ירושה בחץ שבקצהו משולש חלול, המצביע מהמחלקה היורשת אל המחלקה שממנה היא יורשת, ולסמן הכלה בקו שבקצהו מעוין מלא, הצמוד למחלקה המכילה.

(2) כתבו את כותרות המחלקות ואת התכונות שלהן. הניחו שבכל מחלקה הוגדרו פעולות **get** ו-**set** ופעולות בונות.

ב. כתבו במחלקה **Exhibition** פעולה ששמה **totalWeight**, המחזירה את סכום המשקלים של כל הפסלים בתערוכה. אם אין בתערוכה פסלים, הפעולה תחזיר 0.

ג. מערך **arrL** "מסודר" הוא מערך שבו מופיעות קודם ההשאלות של הציורים, אחר כך ההשאלות של הפסלים, ואחר כך ההשאלות של הצילומים.

כתבו במחלקה **Exhibition** פעולה ששמה **remove**, המקבלת שם של מוצג - **title**.

הניחו שלפני תחילת הפעולה המערך "מסודר", ושכל מוצג בתערוכה הוא ציור, פסל או צילום.

אם אין בתערוכה מוצג ששמו title , או שהמוצג ששמו title הוא המוצג **היחיד** מסוגו בתערוכה (למשל, הפסל היחיד בתערוכה), הפעולה תחזיר false ולא תשנה את המערך. אחרת, הפעולה תוציא מהמערך את ההשאלה של המוצג באופן שבו המערך ימשיך להיות "מסודר" (וההשאלות יופיעו ברצף בתחילת המערך), ותחזיר true .

#נוסח C

במוזיאון "**בית האוצרות**" פותחה מערכת ממוחשבת לניהול תערוכות, ובה המחלקות האלה:
Exhibit - מוצג, **Painting** - ציור, **Sculpture** - פסל, **Photo** - צילום, **Loan** - השאלה של מוצג לתערוכה,

Exhibition - תערוכה.

להלן פירוט תכונות המחלקות:

למחלקה **Exhibit** (מוצג) שתי תכונות: title - שם המוצג (מטיפוס מחרוזת), year - שנת היצירה של המוצג (מטיפוס שלם).

למחלקה **Painting** (ציור) שלוש תכונות: title - שם המוצג (מטיפוס מחרוזת), year - שנת היצירה של המוצג (מטיפוס שלם), technique - טכניקת הציור, למשל "שמן" או "צבעי מים" (מטיפוס מחרוזת).

למחלקה **Sculpture** (פסל) שלוש תכונות: title - שם המוצג (מטיפוס מחרוזת), year - שנת היצירה של המוצג (מטיפוס שלם), weight - משקל הפסל בק"ג (מטיפוס שלם).

למחלקה **Photo** (צילום) שלוש תכונות: title - שם המוצג (מטיפוס מחרוזת), year - שנת היצירה של המוצג (מטיפוס שלם), isColor - האם הצילום צבעוני (מטיפוס בוליאני. אם הצילום צבעוני - true , ואם הוא בשחור-לבן - false).

למחלקה **Loan** (השאלה) שתי תכונות: lender - שם המשאיל (מטיפוס מחרוזת), exhibit - המוצג המושאל (מטיפוס **Exhibit**). בכל השאלה מושאל מוצג אחד בלבד.

למחלקה **Exhibition** (תערוכה) שתי תכונות: arrL - מערך של השאלות (מטיפוס **Loan**), count - כמות ההשאלות השמורות במערך (מטיפוס שלם).

ההשאלות מופיעות ברצף מתחילת המערך ללא null ביניהן, אך ייתכן שהמערך אינו מלא עד הסוף. לכל המוצגים בתערוכה יש שמות שונים.

א. (1) סרטטו תרשים הייררכייה בין כל המחלקות.

יש לסמן ירושה בחץ שבקצהו משולש חלול, המצביע מהמחלקה היורשת אל המחלקה שממנה היא יורשת, ולסמן הכלה בקו שבקצהו מעוין מלא, הצמוד למחלקה המכילה.

(2) כתבו את כותרות המחלקות ואת התכונות שלהן. הניחו שבכל מחלקה הוגדרו פעולות Get ו-Set ופעולות בונות.

ב. כתבו במחלקה **Exhibition** פעולה ששמה TotalWeight , המחזירה את סכום המשקלים של כל הפסלים בתערוכה. אם אין בתערוכה פסלים, הפעולה תחזיר 0.

ג. מערך arrL "מסודר" הוא מערך שבו מופיעות קודם ההשאלות של הציורים, אחר כך ההשאלות של הפסלים, ואחר כך ההשאלות של הצילומים.

כתבו במחלקה **Exhibition** פעולה ששמה Remove , המקבלת שם של מוצג - title .

הניחו שלפני תחילת הפעולה המערך "מסודר", ושכל מוצג בתערוכה הוא ציור, פסל או צילום.

אם אין בתערוכה מוצג ששמו title , או שהמוצג ששמו title הוא המוצג היחיד מסוגו בתערוכה (למשל, הפסל היחיד בתערוכה), הפעולה תחזיר false ולא תשנה את המערך. אחרת, הפעולה תוציא מהמערך את ההשאלה של המוצג באופן שבו המערך ימשיך להיות "מסודר" (וההשאלות יופיעו ברצף בתחילת המערך), ותחזיר true .

11. (25 נקודות)

נוסח Java

לפניכם המחלקות Card (כרטיס מועדון) ו-GoldCard (כרטיס זהב):

Java

```
public class Card
{
    private static int counter = 0;
    protected String owner;
    protected int points;

    public Card (String owner) {
        this.owner = owner;
        this.points = 10;
        counter++;
        System.out.println ("Card " + owner);
    }
    public Card (String owner, int points) {
        this.owner = owner;
        this.points = points;
        counter++;
        System.out.println ("Card " + owner + " " + points);
    }
    public static int getCounter() {
        return counter;
    }
    public int getPoints() {
        return points;
    }
    public int value() {
        return this.points * 2;
    }
    public void merge (Card other) {
        this.points += other.points;
        other.points = 0;
        System.out.println (this.owner + ": " + this.points);
    }
}

public class GoldCard extends Card
{
    private int bonus;

    public GoldCard (String owner, int bonus) {
        super (owner);
        this.bonus = bonus;
        System.out.println ("Gold " + bonus);
    }
    public int getBonus() {
        return bonus;
    }
    public int value() {
        return super.value() + this.bonus;
    }
    public void merge (Card other) {
        super.merge (other);
        if (other instanceof GoldCard) {
            this.bonus += ((GoldCard)other).bonus;
            ((GoldCard)other).bonus = 0;
        }
    }
}
```

```

    }
    System.out.println ("bonus: " + this.bonus);
}
}

```

נתונה המחלקה Tester :

Java

```

public class Tester
{
    public static void main(String[] args)
    {
        Card c1 = new Card ("Dan");
        Card c2 = new Card ("Noa", 30);
        Card c3 = new GoldCard ("Tal", 5);
        GoldCard g1 = new GoldCard ("Ori", 8);
        GoldCard g2 = new GoldCard ("Lia", 3);
        // ***
    }
}

```

א. ציירו את העצמים שנוצרו בפעולה main , וכתבו את הפלט של הפעולה.

ב. הציבו כל אחת מן הפקודות 1-10 שלהלן בפעולה main במקום המצוין לעיל ב- ' .

כתבו במחברת את מספר הפקודה וציינו אם הקוד תקין או לא תקין. אם הקוד תקין - כתבו את הפלט (אין צורך לחזור על הפלט של הפעולות הבונות מסעיף א), ואם הוא אינו תקין - הסבירו מדוע.

הערה: אין קשר בין הפקודות. כלומר, יש להתייחס לכל פקודה כאילו היא היחידה בפעולה main .

1. System.out.println ("Cards: " + GoldCard.getCounter());
2. System.out.println (c3.value());
3. System.out.println (c3.getBonus());
4. System.out.println (((GoldCard)c3).getBonus());
5. g2 = c2;
6. GoldCard g3 = (GoldCard)c2;
7. c2.merge (g1);
8. g1.merge (g2);
9. ((Card)g1).merge (c3);
10. Card c4 = new GoldCard ("Ron");

נוסח #C

לפניכם המחלקות Card (כרטיס מועדון) ו-GoldCard (כרטיס זהב):

C#

```

public class Card
{
    private static int counter = 0;
    protected string owner;
    protected int points;

    public Card (string owner) {
        this.owner = owner;
        this.points = 10;
        counter++;
        Console.WriteLine ("Card " + owner);
    }
    public Card (string owner, int points) {
        this.owner = owner;
        this.points = points;
        counter++;
        Console.WriteLine ("Card " + owner + " " + points);
    }
    public static int GetCounter() {
        return counter;
    }
    public int GetPoints() {
        return points;
    }
    public virtual int Value() {
        return this.points * 2;
    }
    public virtual void Merge (Card other) {
        this.points += other.points;
        other.points = 0;
        Console.WriteLine (this.owner + ": " + this.points);
    }
}

public class GoldCard : Card
{
    private int bonus;

    public GoldCard (string owner, int bonus) : base (owner) {
        this.bonus = bonus;
        Console.WriteLine ("Gold " + bonus);
    }
    public int GetBonus() {
        return bonus;
    }
    public override int Value() {
        return base.Value() + this.bonus;
    }
    public override void Merge (Card other) {
        base.Merge (other);
        if (other is GoldCard) {
            this.bonus += ((GoldCard)other).bonus;
            ((GoldCard)other).bonus = 0;
        }
    }
}

```

```
        Console.WriteLine ("bonus: " + this.bonus);
    }
}
```

נתונה המחלקה Tester :

```
C#
public class Tester
{
    public static void Main(string[] args)
    {
        Card c1 = new Card ("Dan");
        Card c2 = new Card ("Noa", 30);
        Card c3 = new GoldCard ("Tal", 5);
        GoldCard g1 = new GoldCard ("Ori", 8);
        GoldCard g2 = new GoldCard ("Lia", 3);
        // ***
    }
}
```

א. ציירו את העצמים שנוצרו בפעולה Main , וכתבו את הפלט של הפעולה.

ב. הציבו כל אחת מן הפקודות 1-10 שלהלן בפעולה Main במקום המצוין לעיל ב- ' .

כתבו במחברת את מספר הפקודה וציינו אם הקוד תקין או לא תקין. אם הקוד תקין - כתבו את הפלט (אין צורך לחזור על הפלט של הפעולות הבונות מסעיף א), ואם הוא אינו תקין - הסבירו מדוע.

הערה: אין קשר בין הפקודות. כלומר, יש להתייחס לכל פקודה כאילו היא היחידה בפעולה Main .

```
1. Console.WriteLine ("Cards: " + GoldCard.GetCounter());
2. Console.WriteLine (c3.Value());
3. Console.WriteLine (c3.GetBonus());
4. Console.WriteLine (((GoldCard)c3).GetBonus());
5. g2 = c2;
6. GoldCard g3 = (GoldCard)c2;
7. c2.Merge (g1);
8. g1.Merge (g2);
9. ((Card)g1).Merge (c3);
10. Card c4 = new GoldCard ("Ron");
```

12. (25 נקודות)

נוסח Java

לשאלה זו שני נוסחים: בשפת Java ובשפת #C .

לפותרים בשפת Java

נתונות המחלקות K , L , M , N :

Java

```
public class K {
    private int v;
    public K (int v) { this.v = v; }
    public int getV() { return v; }
    public String act (K other) {
        System.out.println ("K1");
        return "" + (v + other.getV());
    }
}

-----

public class L extends K {
    public L (int v) { super (v + 2); }
    public String act (L other) {
        System.out.println ("L1");
        return "" + (getV() * other.getV());
    }
    public String act (K other) {
        System.out.println ("L2");
        return "L" + super.act (other);
    }
}

-----

public class M extends L {
    public M (int v) { super (v * 2); }
    public String act (int n) {
        System.out.println ("M1");
        return "" + (getV() - n);
    }
}

-----

public class N extends M {
    public N (int v) { super (v); }
    public String act (L other) {
        System.out.println ("N1");
        return act (other.getV());
    }
}
```

א. סרטטו את תרשים ההיררכיה של המחלקות K , L , M , N . יש לסמן ירושה באמצעות החץ <

. ---

ב. (1) נתונה המחלקה Tester :

Java

```
public class Tester {
    public static void main (String[] args) {
        K k = new K (3);
        L l = new L (1);
        M m = new M (2);
        N n = new N (1);
        K kl = new L (5);
```

```

        L ln = new N (3);
        /**
    }
}

```

סרטטו את העצמים שנוצרו בפעולה main .

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

Java

```

System.out.println (kl.act(1));           //(1)
System.out.println (l.act(ln));           //(2)
System.out.println (ln.act(m));           //(3)
System.out.println (m.act(k));           //(4)
System.out.println (((M)ln).act(5));      //(5)

```

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

נוסח #C

לשאלה זו שני נוסחים: בשפת Java ובשפת #C.

לפותרים בשפת #C

נתונות המחלקות K, L, M, N:

```

public class K {
    private int v;
    public K (int v) { this.v = v; }
    public int GetV() { return v; }
    public virtual string Act (K other) {
        Console.WriteLine ("K1");
        return "" + (v + other.GetV());
    }
}
-----
public class L : K {
    public L (int v) : base (v + 2) { }
    public virtual string Act (L other) {
        Console.WriteLine ("L1");
        return "" + (GetV() * other.GetV());
    }
    public override string Act (K other) {
        Console.WriteLine ("L2");
        return "L" + base.Act (other);
    }
}
-----
public class M : L {
    public M (int v) : base (v * 2) { }
    public string Act (int n) {
        Console.WriteLine ("M1");
        return "" + (GetV() - n);
    }
}
-----
public class N : M {
    public N (int v) : base (v) { }
    public override string Act (L other) {
        Console.WriteLine ("N1");
        return Act (other.GetV());
    }
}

```

א. סרטטו את תרשימים ההיררכייה של המחלקות K, L, M, N. יש לסמן ירושה באמצעות החץ <

ב. (1) נתונה המחלקה Tester:

```

public class Tester {
    public static void Main (string[] args) {
        K k = new K (3);
        L l = new L (1);
        M m = new M (2);
        N n = new N (1);
        K kl = new L (5);
    }
}

```

```

        L ln = new N (3);
        /**
    }
}

```

סרטטו את העצמים שנוצרו בפעולה Main .

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

```

C#
Console.WriteLine (kl.Act(1));           //(1)
Console.WriteLine (l.Act(ln));           //(2)
Console.WriteLine (ln.Act(m));           //(3)
Console.WriteLine (m.Act(k));           //(4)
Console.WriteLine (((M)ln).Act(5));      //(5)

```

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

כל הבגרויות במדעי המחשב עם פתרונות מלאים: www.guyai.co.il/bagrut/computer-science