

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 7
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)

נתונה המחלקה **Update** - עדכון מחיר של מוצר, ולה שתי תכונות:

· **product** - קוד המוצר, מטיפוס שלם

· **price** - המחיר החדש של המוצר (בשקלים), מטיפוס שלם

הניחו שיש פעולות **get/Set** ו- **set/Set** לתכונות המחלקה, ופעולה בונה המקבלת ערכים בעבור תכונות המחלקה.

במכולת השכונתית "פינת הכרם" נשמרים כל עדכוני המחירים של יום אחד בתור - **qUpd** מטיפוס **Update**, **לפי סדר הזמן** שבו נעשו: העדכון המוקדם ביותר נמצא בראש התור.

הערות:

· מחיר של מוצר יכול להתעדכן כמה פעמים באותו היום, ולכן אותו מוצר יכול להופיע בתור כמה פעמים, במקומות שונים.

· המחיר **התקף** של מוצר בסוף היום הוא המחיר שבעדכון **האחרון** שלו (הקרוב ביותר לסוף התור).

למשל, אם בתור יש (משמאל - ראש התור): (215,40), (308,12), (215,36), (122,9), (215,44) - המחיר התקף של מוצר 215 בסוף היום הוא 44.

א. (1) ממשו את הפעולה שלפניכם:

```
Java - public static Queue<Update> latestPrices (Queue<Update> qUpd)
```

```
#C - public static Queue<Update> LatestPrices (Queue<Update> qUpd)
```

הפעולה מקבלת תור עדכונים - **qUpd**, ומחזירה תור **חדש** של עצמים מטיפוס **Update**, ובו כל מוצר מופיע פעם אחת בלבד - עם המחיר התקף שלו בסוף היום.

בעבור התור שבדוגמה, הפעולה תחזיר תור ובו: (215,44), (308,12), (122,9).

הערות:

· אין חשיבות לסדר העצמים בתור המוחזר.

· **אין להשתמש בסעיף זה במערך או ברשימה מקושרת** (או בשום מבנה נתונים אחר פרט לתור). פתרון הכולל שימוש בהם לא יזוכה בנקודות.

· אפשר לשנות את התור שהתקבל.

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

ב. (1) בנוסף לתור העדכונים של היום, נתון תור **yesterday** מטיפוס **Update** - מחירון סוף היום הקודם: כל מוצר של המכולת מופיע בו **פעם אחת בדיוק**, עם המחיר שלו אתמול. כל מוצר שמופיע בתור העדכונים של היום מופיע גם ב-**yesterday**.

ממשו את הפעולה שלפניכם:

```
Java - public static Queue<Integer> priceRise (Queue<Update> qUpd, Queue<Update> yesterday)
```

```
public static Queue<int> PriceRise (Queue<Update> qUpd, Queue<Update> -#C
                                     yesterday)
```

הפעולה מחזירה תור מטיפוס **שלם** ובו הקודים של כל המוצרים שהמחיר התקף שלהם בסוף היום **גבוה** מן המחיר שלהם ב-yesterday. כל מוצר יופיע פעם אחת לכל היותר (ללא חשיבות לסדר).

דוגמה: $q_{\text{Upd}} = (102, 15), (101, 28), \neg \text{yesterday} = (101, 30), (102, 12), (103, 45), (104, 8)$
 $\leftarrow 103, 33 \leftarrow 101, 11 \leftarrow 102$: המחירים התקפים בסוף היום: $(102, 11), (103, 50), (101, 33)$
50. הפעולה תחזיר תור ובו 101 ו-103 (מוצר 102 הוזל מ-12 ל-11, ומוצר 104 לא עודכן היום).
אפשר להשתמש בפעולה שכתבתם בסעיף א.

הערות:

- **אין להשתמש בסעיף זה במערך או ברשימה מקושרת** (או בשום מבנה נתונים אחר פרט לתור). פתרון הכולל שימוש בהם לא יזוכה בנקודות.
- אפשר לשנות את התור qUpd. בסיום הפעולה התור yesterday צריך להיות **כפי שהיה** בתחילתה.

(2) מהי סיבוכיות זמן הריצה של הפעולה? נמקו את תשובתכם. סמנו ב- n את מספר העדכונים ב- q וב- m את מספר המוצרים ב- $yesterday$.

2. (25 נקודות)

כדי לחסוך בזיכרון, מחרוזת שיש בה תווים חוזרים ברצף נשמרת בשתי שרשרות של חוליות:

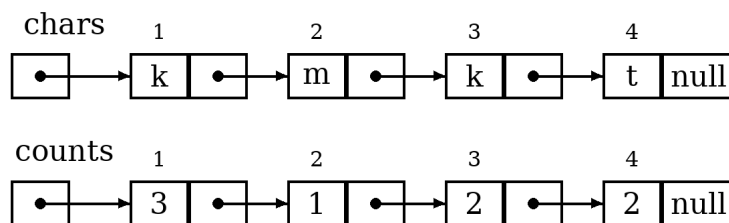
chars - הפניה לשרשרת חוליות מטיפוס תו (char).

counts - הפניה לשרשרת חוליות של מספרים שלמים **חיוביים**.

הערה: כמות החוליות בשתי השרשרות שווה.

המספר במיקום 1 בשרשרת counts מתאר כמה פעמים ברצף מופיע התו שבמיקום 1 בשרשרת chars, המספר במיקום 2 מתאר כמה פעמים ברצף מופיע התו שבמיקום 2, וכן הלאה עד סוף שתי השרשרות. המחרוזת המיוצגת מתקבלת מחיבור כל הרצפים לפי הסדר.

דוגמה: שתי השרשרות שלפניכם מייצגות את המחרוזת "kkkmmkkttt" (שלוש פעמים k, פעם אחת m, פעמיים k ופעמיים t). אורך המחרוזת הוא 8.



א. ממשו את הפעולה שלפניכם:

[illegible]

```
public static char CharAt (Node<char> chars, Node<int> counts, int pos) - #C
```

הפעולה תחזיר את התו שנמצא במקום pos במחרוזת המיוצגת על ידי שתי השרשרות. המקומות במחרוזת ממוספרים החל מ-1.

הניחו ש- pos גדול מ-0. אין לשנות את השרשרות שהתקבלו. **אין** לבנות את המחרוזת המיוצגת.

הערה: אם pos גדול מאורך המחרוזת המיוצגת, הפעולה תחזיר את התו '#'.
דוגמה: בעבור שתי השרשרות בדוגמה שלעיל ו- pos = 4, הפעולה תחזיר 'm'.
דוגמה נוספת: בעבור אותן שתי שרשרות ו- pos = 7, הפעולה תחזיר 't'.
דוגמה נוספת: בעבור אותן שתי שרשרות ו- pos = 20, הפעולה תחזיר '#', כי אורך המחרוזת הוא 8 בלבד.

ב. ממשו את הפעולה שלפניכם:

```
public static boolean isPalindrome (Node<Character> chars, Node<Integer> counts)
```

```
public static bool IsPalindrome (Node<char> chars, Node<int> counts) - #C
```

הפעולה תחזיר true אם המחרוזת המיוצגת על ידי שתי השרשרות היא **פלינדרום** (נקראת באותו אופן משמאל לימין ומימין לשמאל), אחרת הפעולה תחזיר false.

אפשר להשתמש בפעולה שכתבתם בסעיף א. אין לשנות את השרשרות שהתקבלו. **אין** לבנות את המחרוזת המיוצגת. הניחו שהשרשרות אינן ריקות.

שימו לב: השרשרות עצמן אינן חייבות להיות סימטריות.

דוגמה: בעבור chars = s, a, s ו- counts = 2, 1, 2 (המחרוזת "ssass") הפעולה תחזיר true.

דוגמה נוספת: בעבור chars = b, c, b, b ו- counts = 2, 1, 1, 1 (המחרוזת "bbcb") הפעולה תחזיר true.

דוגמה נוספת: בעבור שתי השרשרות בדוגמה שלעיל (המחרוזת "kkkmkkt") הפעולה תחזיר false.

3. (25 נקודות)

בשאלה זו אפשר להשתמש בפעולה החיצונית isLeaf / IsLeaf שלהלן בלי לממש אותה.

דוגמאות	תיאור הפעולה	כותרת הפעולה
- בעבור צומת שאין לו בנים, הפעולה תחזיר true. - בעבור צומת שיש לו בן ימני בלבד, הפעולה תחזיר false.	הפעולה מקבלת הפניה לצומת node שאינו null. הפעולה מחזירה true אם node הוא עלה (אין לו בן שמאלי ואין לו בן ימני), אחרת היא מחזירה false.	Java: <code>public static boolean isLeaf (BinNode<Integer> node)</code> בשפת C: <code>public static bool IsLeaf (BinNode<int> node)</code>

ממשו את הפעולה החיצונית שלהלן:

Java - `public static int countPaths (BinNode<Integer> tree, int sum)`

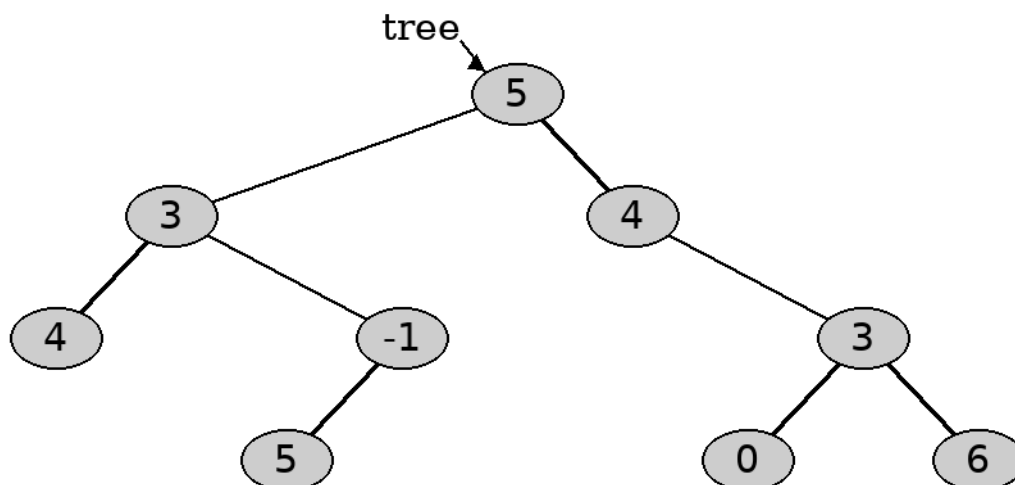
#C - `public static int CountPaths (BinNode<int> tree, int sum)`

הפעולה מקבלת הפניה לעץ בינארי של מספרים שלמים - tree שאינו null, ומספר שלם - sum. הפעולה תחזיר את מספר המסלולים בעץ שמתחילים **בשורש** העץ, מסתיימים **בעלה**, וסכום הערכים בצמתים שלהם שווה ל- sum.

הערה: בעץ יכולים להופיע גם המספר 0 ומספרים שליליים.

דוגמה:

בעבור העץ הנתון ו- sum = 12 הפעולה תחזיר 3.



הסבר: בעץ יש שלושה מסלולים מהשורש לעלה שסכומם 12: המסלול 5, 3, -1; המסלול 5, 3, 4; והמסלול 5, 4, 0.

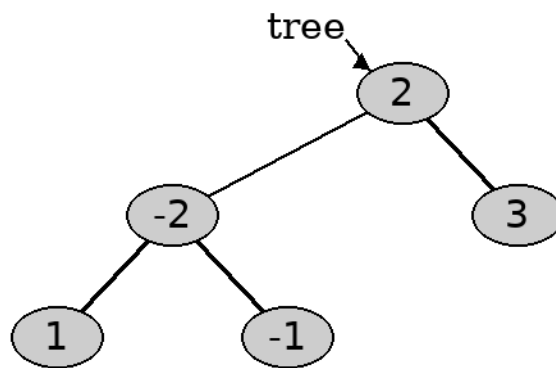
דוגמה:

בעבור אותו העץ ו- sum = 9 הפעולה תחזיר 0.

הסבר: סכום הערכים במסלול 5, 4 הוא 9, אבל המסלול אינו מסתיים בעלה (לצומת 4 יש בן). סכומי המסלולים מהשורש לעלים הם 12, 12, 12 ו- 18.

דוגמה:

בעבור העץ הנתון ו- sum = 1 הפעולה תחזיר 1.



הסבר: המסלול היחיד מהשורש לעלה שסכומו 1 הוא 2, 1, -2.
ממשו את הפעולה countPaths / CountPaths המתוארת לעיל.

פרק שני (50 נקודות) - אלגוריתמים

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

4. (25 נקודות)

לפניכם שני סעיפים, א-ב, שאין קשר ביניהם. ענו על שני הסעיפים.

א. לפניכם שבע טענות (1)-(7). בחרו בחמש מהן, וכתבו את מספריהן. ציינו בנוגע לכל טענה שבחרתם

אם היא נכונה או לא נכונה. אם הטענה נכונה - נמקו מדוע, ואם היא לא נכונה - הביאו דוגמה

נגדית מגרף שיש בו ארבעה קודקודים לפחות.

(1) נתון גרף G קשיר שאינו מכוון, ומריצים עליו BFS מקודקוד s . אם $d(v)$ הוא המרחק של

הקודקוד v מ- s שמחשב האלגוריתם, אז לכל קשת (u, v) בגרף מתקיים: $|d(u) - d(v)| \leq 1$.

(2) בגרף מכוון ממושקל שיש בו קשתות במשקל שלילי, אבל אין בו מעגל שלילי, האלגוריתם של

דייקסטרה מחשב תמיד נכון את המרחקים הקצרים (הקלים) מקודקוד המקור לכל הקודקודים.

(3) נתון גרף ממושקל קשיר לא מכוון. אם מוסיפים מספר קבוע $c > 0$ למשקל של כל קשת בגרף, כל

עץ פורש מינימלי של הגרף המקורי הוא גם עץ פורש מינימלי של הגרף החדש.

(4) נתון גרף ממושקל שבו המשקל של כל קשת חיובי. אם מוסיפים מספר קבוע $c > 0$ למשקל של

כל קשת בגרף, כל מסלול קצר (קל) מקודקוד s לקודקוד t בגרף המקורי הוא גם מסלול קצר (קל)

מ- s ל- t בגרף החדש.

(5) בכל גרף מכוון חסר מעגלים יש לפחות קודקוד אחד שדרגת הכניסה שלו היא 0.

(6) נתון גרף קשיר לא מכוון. הגובה של כל עץ DFS שמתקבל מסריקה מקודקוד s גדול מן הגובה

של עץ ה-BFS מ- s או שווה לו. (גובה של עץ הוא מספר הקשתות במסלול הארוך ביותר מן השורש

לעלה.)

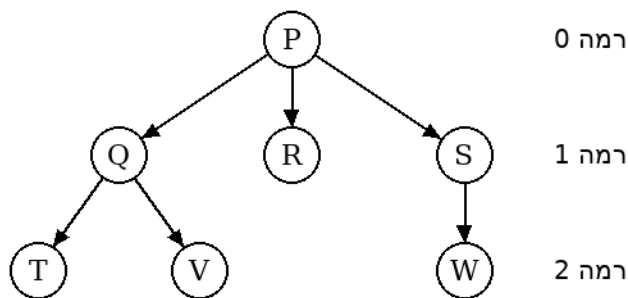
(7) אם בגרף מכוון ממושקל יש קשת שמשקלה שלילי, האלגוריתם של בלמן-פורד יודיע שיש בגרף

מעגל שלילי.

ב. עץ BFS מקודקוד s הוא העץ שמתקבל מן הקשתות שדרכן התגלה כל קודקוד בסריקת BFS מ- s .

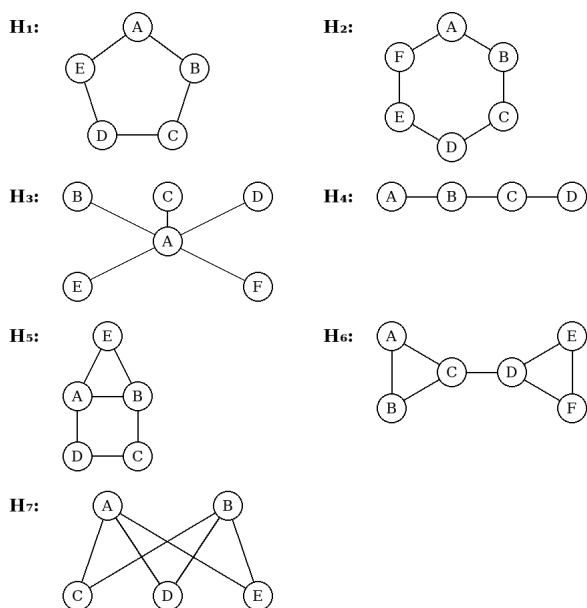
הרמה של קודקוד בעץ היא מספר הקשתות בעץ מן השורש s עד הקודקוד (השורש נמצא ברמה 0).

דוגמה לעץ BFS שבו כל הקודקודים נמצאים ברמה 2 לכל היותר:



גרף קשיר לא מכוון מכונה "קריב" אם בכל סריקת BFS מכל קודקוד בגרף מתקבל עץ BFS שבו כל הקודקודים נמצאים ברמה 2 לכל היותר.

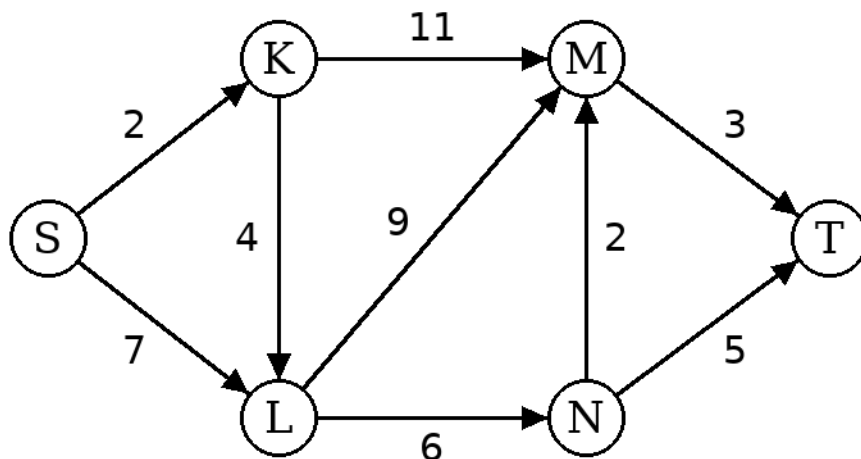
לפניכם שבעה גרפים H_1-H_7 . בחרו בארבעה מהם, וכתבו בנוגע לכל גרף שבחרתם אם הוא גרף "קריב" או לא. אם כתבתם שלא - הציגו סריקת BFS (קודקוד התחלה והרמה של כל קודקוד) שבעבורה מתקבל עץ שיש בו קודקוד ברמה 3 או יותר.



5. (25 נקודות)

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

א. (1) לפניכם גרף מכוון ממושקל G_1 :



(1) איזה אלגוריתם מחזיר את המסלולים הקצרים (הקלים) מקודקוד מקור לכל קודקודי הגרף, בגרף ממושקל **כלשהו** שכל משקליו **אי-שליליים**? מהי סיבוכיות זמן הריצה של האלגוריתם? נמקו את תשובתכם.

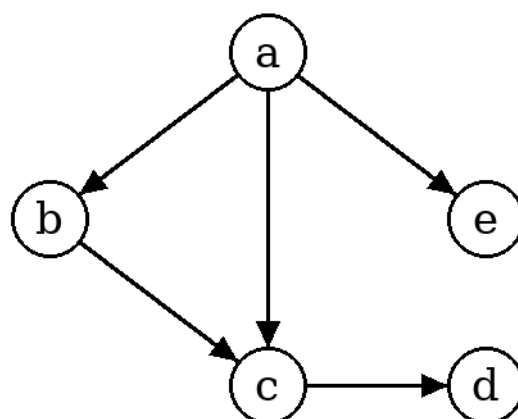
נוסף על כך, הריצו את האלגוריתם שכתבתם, בעבור הגרף הנתון G , מקודקוד S לקודקוד T . בצעו מעקב שלב אחר שלב: בכל שלב רשמו איזה קודקוד נבחר, ואת ערכי d (המרחק הידוע) ו- π (הקודקוד הקודם) של הקודקודים. רשמו את המסלול שהתקבל ואת אורכו (סכום משקליו).

(2) עבור גרף G ממושקל **כלשהו** שכל משקליו מספרים שלמים **חיוביים**, הרצנו את האלגוריתם המוצא את המסלול הקצר (הקל) בין שני קודקודים s ו- t , וקיבלנו מסלול P .

- האם P יישאר בהכרח מסלול קצר ביותר בין s ל- t אם משקל של קשת **אחת** בגרף, **שאינה** נמצאת על המסלול P , **יגדל** ב- x ($0 < x$)? נמקו את תשובתכם.

- האם P יישאר בהכרח מסלול קצר ביותר בין s ל- t אם משקל **כל אחת** מן הקשתות בגרף **יועלה בריבוע** (כל משקל w יוחלף ב- w^2)? נמקו את תשובתכם.

ב. (1) 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 שבארבע סריקות DFS מקודקוד a, יתקבלו ארבעה עצים שהגבהים שלהם שונים זה מזה. סרטטו את הגרף לאחר ההוספה, והציגו את ארבע הסריקות.

6. (25 נקודות)

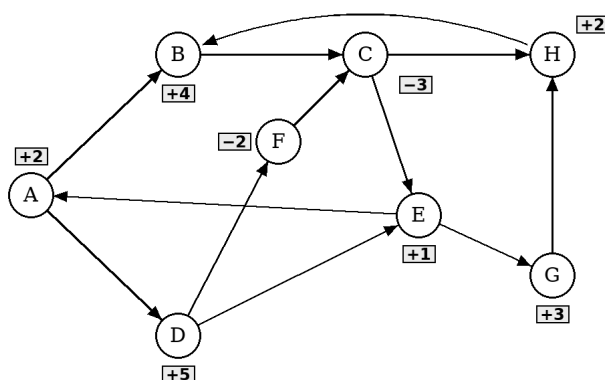
במשחק מחשב יש מבוכ של חדרים. בין חלק מן החדרים יש דלתות חד-כיווניות: אפשר לעבור בדלת רק בכיוון החץ.

לכל חדר יש ערך (מספר שלם):

· ערך חיובי הוא דמי מעבר: שחקן שנכנס לחדר משלם מספר מטבעות השווה לערך.

· ערך שלילי מציין אוצר: שחקן שנכנס לחדר מקבל מספר מטבעות השווה לערך המוחלט.

המחיר של מסלול הוא סכום הערכים של החדרים שהשחקן נכנס אליהם במסלול. המחיר אינו כולל את ערך החדר שבו השחקן מתחיל את מסלולו, אך הוא כולל את ערך החדר שבו השחקן מסיים את מסלולו. המסלול הזול ביותר בין שני חדרים הוא המסלול שהמחיר שלו הקטן ביותר. ידוע שבמבוכ אין מסלול מעגלי (מסלול שחוזר אל החדר שממנו יצא) שהמחיר שלו שלילי. לפניכם המבוכ MAZE. כל עיגול מייצג חדר, והמספר שבמלבן ליד כל חדר הוא הערך שלו:



דוגמה: המחיר של המסלול $A \rightarrow B \rightarrow C \rightarrow E$ הוא $2 = 1 + (-3) + 4$.

א. במבוכ MAZE, מהו המסלול הזול ביותר מחדר A לחדר H? כתבו את שמות החדרים במסלול זה לפי הסדר, ואת המחיר שלו (אין צורך לבצע מעקב).

ב. (1) כתבו אלגוריתם המוצא בכל מבוכ (שמקיים את התנאים שלעיל) את המסלול הזול ביותר מחדר K_1 לחדר K_2 .

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

(2) סרטטו את הגרף המייצג את המבוכ MAZE הנתון לעיל, באופן שיתאים לאלגוריתם שכתבתם.

פרק שני (50 נקודות) - מודלים חישוביים

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

7. (25 נקודות)

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

- א.** (1) לפניכם שתי טענות (1)-(2) בנוגע לשפות שמעל הא"ב $\{a, b\}$.
 בעבור כל טענה, ציינו אם היא נכונה או לא נכונה. אם הטענה נכונה - נמקו מדוע,
 ואם היא לא נכונה - הביאו דוגמה נגדית. אין קשר בין הטענות.
 (1) אם L_2 היא שפה רגולרית ו- $L_1 \subseteq L_2$, בהכרח L_1 היא שפה רגולרית.
 (2) לכל שתי שפות L_1, L_2 מתקיים: $L^R = L_1^R \cap L_2^R(L_1 \cap L_2)$.
 L^R מציין את השפה שמילותיה הן היפוכי המילים של L .
- ב.** (1) נתונה השפה L מעל הא"ב $\{a, b, c\}$:
 $\{w \mid w = aa \text{ הרצף } aa \text{ אינו מופיע ב- } w, \#c(w) \text{ מתחלק ב- } 3 \text{ ללא שארית}\}$
 $\#c(w)$ מציין את מספר המופעים של c במילה w .
 דוגמה למילה בשפה L היא $abcbcac$, כי מספר המופעים של האות c הוא 3, והרצף aa אינו מופיע במילה.
 (1) לפניכם חמש מילים. בנוגע לכל אחת מהן ציינו אם המילה שייכת לשפה L או לא. נמקו את תשובתכם.
 $ccc, baac, \varepsilon, acacbc, cbc$.
 (2) בנו אוטומט סופי דטרמיניסטי שאינו מלא המקבל את השפה L .

8. (25 נקודות)

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

- א.** (1) לפניכם שש שפות מעל הא"ב $\{a, b\}$:
 Σ^* - מציין את שפת כל המילים מעל הא"ב $\{a, b\}$.
 L^R - מציין את השפה שמילותיה הן היפוכי המילים של L .

$L_4 = \{ab, b\}$	$L_1 = \{\varepsilon\}$
$L_5 = \{\varepsilon, a, bb\}$	$L_2 = \emptyset$
$L_6 = \{ba, ab, aab\}$	$L_3 = \Sigma$

כתבו את השפה המתקבלת מכל אחת מחמש הפעולות שלהלן:

$$L_4 \cup L_6 \quad (1)$$

$$L_6^R \quad (2)$$

$$L_3 \cap L_6 \quad (3)$$

$$L_2 \cup L_1 \quad (4)$$

$$L_5 \cdot L_4 \quad (5)$$

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

$$L = \{a^{2n} b^m c^{n+m} \mid n, m \geq 0\}$$

בנו אוטומט מחסנית דטרמיניסטי המקבל את השפה L .

9. (25 נקודות)

כתבו מכונת טיורינג המקבלת בתחילת הסרט מספר בינארי הגדול מאפס (בלי אפסים מובילים, כלומר הספרה השמאלית היא 1), ומחזירה את אותו המספר בייצוג **אוני**. 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 ראשונה (המספר 6)

הסרט לפני ההרצה:

⊢	1	1	0	△	△	△	△	△
---	---	---	---	---	---	---	---	---

הסרט לאחר ההרצה:

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

דוגמה שנייה (המספר 11)

הסרט לפני ההרצה:

⊢	1	0	1	1	△	△	△	△	△
---	---	---	---	---	---	---	---	---	---

הסרט לאחר ההרצה:

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

כתבו מכונת טיורינג המקבלת בתחילת הסרט מספר בינארי הגדול מאפס, ומחזירה את אותו המספר בייצוג אוני, כמפורט לעיל.
(25 נקודות)

תכנות מונחה עצמים

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

10. (25 נקודות)

נוסח Java

משתלת "ירוק עד" מעוניינת למכור את מוצריה גם באופן מקוון (אונליין). בשלב הראשון החליטה המשתלה למכור באופן מקוון רק עצים ופרחים.

לשם כך נבנו המחלקות האלה:

- המחלקה **Plant**, המייצגת צמח

- המחלקה **Tree**, המייצגת עץ

- המחלקה **Flower**, המייצגת פרח

Java

```
public class Plant {
    private String code;        // קוד המוצר
    private String name;        // שם הצמח
    private String supplier;    // שם הספק
    private double price;       // מחיר
    private int height;         // גובה בסנטימטרים
}

public class Tree extends Plant {
    private int age;            // גיל העץ בשנים
}

public class Flower extends Plant {
    private String color;       // צבע הפרח
    private boolean perennial; // האם הפרח רב־שנתי
}
```

הניחו שיש פעולות get ו־set בעבור התכונות בכל המחלקות.

א. (1) המחלקה **Delivery** מייצגת משלוח ללקוח. למחלקה שלוש תכונות:

. address - כתובת המשלוח

. plants - מערך המכיל גם עצים וגם פרחים

. count - מספר הצמחים הנוכחי במשלוח

(1) כתבו את כותרת המחלקה **Delivery** ואת התכונות שלה.

(2) סרטטו תרשים הייררכייה המתאר את הקשר בין המחלקות: **Plant**, **Tree**, **Flower**, **Delivery**.

יש לסמן ירושה באמצעות החץ <--- והכלה באמצעות הסימן ◆---.

ב. (1) בשלב השני החליטה המשתלה למכור באופן מקוון גם כלי גינון.

לפניכם המחלקה **Tool** המייצגת כלי גינון:

Java

```
public class Tool {
    private String code;        // קוד המוצר
    private String material;    // החומר שממנו עשוי הכלי
    private String supplier;    // שם הספק
    private double price;       // מחיר
    private int warranty;       // תקופת האחריות בחודשים
}
```

הניחו שיש פעולות get ו־set בעבור תכונות המחלקה.

כדי שהמערך במחלקה **Delivery** יוכל להכיל גם כלי גינון, נעשו הפעולות האלה:

- הוגדרה המחלקה **Item** - מוצר.

- המערך **plants** במחלקה **Delivery** הוחלף במערך **items** מטיפוס **Item**.

- נכתבו במחלקות המתאימות (כולל במחלקה **Tool**) יחסי הורשה חדשים בינן ובין המחלקה **Item**.

- חלק מן התכונות שהופיעו עד עכשיו בכמה מן המחלקות האחרות (כולל המחלקה **Tool**) הועברו למחלקה **Item**, בהתאם לעקרונות תכנות מונחה עצמים (מעבר לתכונות אלו אין תכונות נוספות במחלקה **Item**).

(1) סרטטו תרשים הייררכייה מעודכן המתאר את הקשר בין המחלקות: **Plant**, **Tree**, **Flower**, **Tool**, **Item**, **Delivery**.

יש לסמן ירושה באמצעות החץ <--- והכלה באמצעות הסימן ◆---.

(2) כתבו את כותרת המחלקה **Item** ותכונותיה.

(3) כתבו מחדש את המחלקות שחל בהן שינוי בעקבות ההוספה של המחלקה **Item**.

שימו לב: יש להתייחס גם ליחסי הורשה חדשים וגם לשינוי שחל בתכונות המחלקה.

ג. בשלב האחרון החליטה המשתלה על מבצע סוף עונה: ברכישה מקוונת כל המוצרים יימכרו בהנחה של 5%, ובעבור צמחים (עצים ופרחים) תינתן הנחה נוספת של 20% מן המחיר שאחרי ההנחה הראשונה.

נתונה הפעולה הפנימית במחלקה **Item**:

Java

```
public double getSalePrice()
```

זימון הפעולה מחזיר את מחיר המוצר (עץ/פרח/כלי גינון) לאחר ההנחה - בהתאם לכל מוצר.

ממשו את הפעולה במחלקה **Item**, והוסיפו פעולה/ות במחלקה/ות המתאימה/ות כדי לבצע את הנדרש. ציינו בעבור כל פעולה שהוספתם באיזו מחלקה היא נוספה.

אין להשתמש בפעולה **instanceof** ובפעולות המחלקה **Object** **ואין** לשנות את תכונות המחלקות.

נוסח C#

משתלת "ירוק עד" מעוניינת למכור את מוצריה גם באופן מקוון (אונליין). בשלב הראשון החליטה המשתלה למכור באופן מקוון רק עצים ופרחים.
 לשם כך נבנו המחלקות האלה:
 - המחלקה **Plant**, המייצגת צמח
 - המחלקה **Tree**, המייצגת עץ
 - המחלקה **Flower**, המייצגת פרח

C#

```
public class Plant
{
    private string code;        // קוד המוצר
    private string name;        // שם הצמח
    private string supplier;    // שם הספק
    private double price;       // מחיר
    private int height;         // גובה בסנטימטרים
}

public class Tree : Plant
{
    private int age;            // גיל העץ בשנים
}

public class Flower : Plant
{
    private string color;       // צבע הפרח
    private bool perennial;     // האם הפרח רב־שנתי
}
```

הניחו שיש פעולות Get ו-Set בעבור התכונות בכל המחלקות.

א. (1) המחלקה **Delivery** מייצגת משלוח ללקוח. למחלקה שלוש תכונות:

· address - כתובת המשלוח

· plants - מערך המכיל גם עצים וגם פרחים

· count - מספר הצמחים הנוכחי במשלוח

(1) כתבו את כותרת המחלקה **Delivery** ואת התכונות שלה.

(2) סרטטו תרשים הייררכייה המתאר את הקשר בין המחלקות: **Plant**, **Tree**, **Flower**, **Delivery**.

יש לסמן ירושה באמצעות החץ <--- והכלה באמצעות הסימן ◆---.

ב. (1) בשלב השני החליטה המשתלה למכור באופן מקוון גם כלי גינון.

לפניכם המחלקה **Tool** המייצגת כלי גינון:

C#

```
public class Tool
{
    private string code;        // קוד המוצר
    private string material;    // החומר שממנו עשוי הכלי
    private string supplier;    // שם הספק
    private double price;       // מחיר
    private int warranty;       // תקופת האחריות בחודשים
}
```

הניחו שיש פעולות Get ו- Set בעבור תכונות המחלקה.

כדי שהמערך במחלקה Delivery יוכל להכיל גם כלי גינון, נעשו הפעולות האלה:

- הוגדרה המחלקה **Item** - מוצר.

- המערך plants במחלקה Delivery הוחלף במערך items מטיפוס Item.

- נכתבו במחלקות המתאימות (כולל במחלקה Tool) יחסי הורשה חדשים בין ובין המחלקה Item.

- חלק מן התכונות שהופיעו עד עכשיו בכמה מן המחלקות האחרות (כולל המחלקה Tool) הועברו למחלקה Item, בהתאם לעקרונות תכנות מונחה עצמים (מעבר לתכונות אלו אין תכונות נוספות במחלקה Item).

(1) סרטוט תרשים הייררכייה מעודכן המתאר את הקשר בין המחלקות: Plant, Tree, Flower, Tool, Item, Delivery.

יש לסמן ירושה באמצעות החץ <--- והכלה באמצעות הסימן ◆---.

(2) כתבו את כותרת המחלקה Item ותכונותיה.

(3) כתבו מחדש את המחלקות שחל בהן שינוי בעקבות ההוספה של המחלקה Item.

שימו לב: יש להתייחס גם ליחסי הורשה חדשים וגם לשינוי שחל בתכונות המחלקה.

ג. בשלב האחרון החליטה המשתלה על מבצע סוף עונה: ברכישה מקוונת כל המוצרים יימכרו בהנחה של 5%, ובעבור צמחים (עצים ופרחים) תינתן הנחה נוספת של 20% מן המחיר שאחרי ההנחה הראשונה.

נתונה הפעולה הפנימית במחלקה Item:

C#

```
public virtual double GetSalePrice()
```

זימון הפעולה מחזיר את מחיר המוצר (עץ/פרח/כלי גינון) לאחר ההנחה - בהתאם לכל מוצר.

ממשו את הפעולה במחלקה Item, והוסיפו פעולה/ות במחלקה/ות המתאימה/ות כדי לבצע את הנדרש. ציינו בעבור כל פעולה שהוספתם באיזו מחלקה היא נוספה.

אין להשתמש בפעולות is ו- as ובפעולות המחלקה object **ואין** לשנות את תכונות המחלקות.

11. (25 נקודות)

נוסח Java

לפניכם כותרות המחלקות MM, NN והתכונות שלהן.
הפעולות הבונות של המחלקות מסומנות ב- * (תיתכן יותר מפעולה בונה אחת למחלקה).

Java

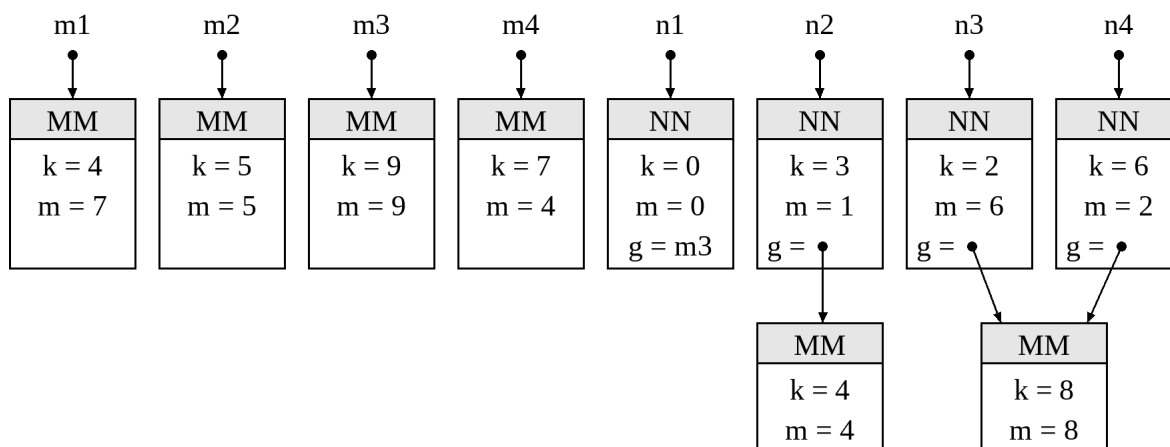
```
public class MM
{
    private int k;
    private int m;
    ***
}
public class NN extends MM
{
    private MM g;
    ***
}
```

הערה: שימו לב - לשתי המחלקות אין פעולות get ו- set.
לפניכם המחלקה Test, הכוללת פעולה ראשית:

Java

```
public class Test {
    public static void main(String[] args) {
        MM m1 = new MM (4, 7);
        MM m2 = new MM (5);
        MM m3 = new MM (9);
        MM m4 = new MM (m1);
        NN n1 = new NN (m3);
        NN n2 = new NN (3, 1);
        NN n3 = new NN (2, 6);
        NN n4 = new NN (n3);
    }
}
```

לפניכם תרשים של עצמים שנוצרו בעקבות הרצת קטע הקוד.
כתבו במחלקות MM ו- NN את הפעולות הבונות הנדרשות כדי לקבל את העצמים שבתרשים.
ציינו בעבור כל אחד מן העצמים שנוצרו את הפעולה הבונה המתאימה לו.
הערה: אין להוסיף פעולות שאינן בונות במחלקות MM ו- NN. פתרון המוסיף פעולות שאינן בונות לא יזוכה בנקודות.



נוסח #C

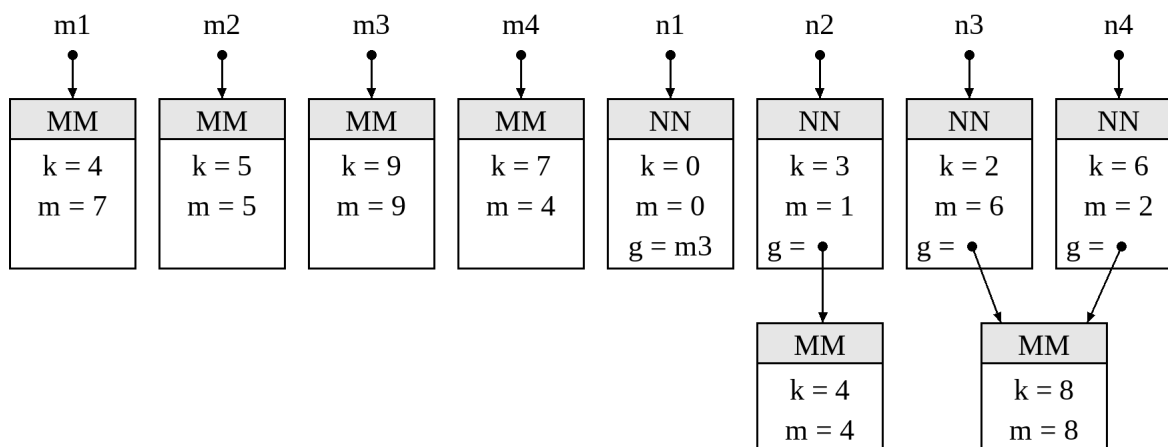
לפניכם כותרות המחלקות MM, NN והתכונות שלהן.
הפעולות הבונות של המחלקות מסומנות ב- * (תיתכן יותר מפעולה בונה אחת למחלקה).

```
public class MM
{
    private int k;
    private int m;
    ***
}
public class NN : MM
{
    private MM g;
    ***
}
```

הערה: שימו לב - לשתי המחלקות אין פעולות Get ו- Set.
לפניכם המחלקה Test, הכוללת פעולה ראשית:

```
public class Test {
    public static void Main(string[] args) {
        MM m1 = new MM (4, 7);
        MM m2 = new MM (5);
        MM m3 = new MM (9);
        MM m4 = new MM (m1);
        NN n1 = new NN (m3);
        NN n2 = new NN (3, 1);
        NN n3 = new NN (2, 6);
        NN n4 = new NN (n3);
    }
}
```

לפניכם תרשים של עצמים שנוצרו בעקבות הרצת קטע הקוד.
כתבו במחלקות MM ו- NN את הפעולות הבונות הנדרשות כדי לקבל את העצמים שבתרשים.
ציינו בעבור כל אחד מן העצמים שנוצרו את הפעולה הבונה המתאימה לו.
הערה: אין להוסיף פעולות שאינן בונות במחלקות MM ו- NN. פתרון המוסיף פעולות שאינן בונות לא יזוכה בנקודות.



12. (25 נקודות)**נוסח Java**

לשאלה זו שני נוסחים: בשפת Java ובשפת C#.

לפותרים בשפת Java

נתונות המחלקות P, Q, R, S:

Java

```
public class P {
    private int k;
    public P (int k) { this.k = k; }
    public int getK() { return k; }
    public int show (int x) {
        System.out.println ("P-int");
        return k + x;
    }
}

-----

public class Q extends P {
    public Q (int k) { super (k * 3); }
    public int show (P other) {
        System.out.println ("Q-P");
        return getK() - other.getK();
    }
    public int show (int x) {
        System.out.println ("Q-int");
        return super.show (x + 1);
    }
}

-----

public class R extends Q {
    public R (int k) { super (k - 1); }
    public int show (Q other) {
        System.out.println ("R-Q");
        return other.getK();
    }
}

-----

public class S extends P {
    public S (int k) { super (k + 4); }
    public int show (P other) {
        System.out.println ("S-P");
        return other.show (getK());
    }
}
```

א. סרטטו את תרשימי ההיררכיה של המחלקות P, Q, R, S. יש לסמן ירושה באמצעות החץ <---

ב. (1) נתונה המחלקה Tester:

Java

```
public class Tester {
    public static void main (String[] args) {
        P p = new P (2);
        Q q = new Q (1);
    }
}
```

```

        R r = new R (4);
        S s = new S (3);
        P pq = new Q (2);
        Q qr = new R (5);
        /**
    }
}
```

סרטטו את העצמים שנוצרו בפעולה main .

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

Java

```

System.out.println (pq.show(1));           //(1)
System.out.println (qr.show(r));           //(2)
System.out.println (r.show(qr));           //(3)
System.out.println (((Q)pq).show(s));      //(4)
System.out.println (s.show(pq));           //(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 P, Q, R, S:

```

public class P {
    private int k;
    public P (int k) { this.k = k; }
    public int GetK() { return k; }
    public virtual int Show (int x) {
        Console.WriteLine ("P-int");
        return k + x;
    }
}
-----
public class Q : P {
    public Q (int k) : base (k * 3) { }
    public int Show (P other) {
        Console.WriteLine ("Q-P");
        return GetK() - other.GetK();
    }
    public override int Show (int x) {
        Console.WriteLine ("Q-int");
        return base.Show (x + 1);
    }
}
-----
public class R : Q {
    public R (int k) : base (k - 1) { }
    public int Show (Q other) {
        Console.WriteLine ("R-Q");
        return other.GetK();
    }
}
-----
public class S : P {
    public S (int k) : base (k + 4) { }
    public int Show (P other) {
        Console.WriteLine ("S-P");
        return other.Show (GetK());
    }
}

```

א. סרטטו את תרשימים ההיררכייה של המחלקות P, Q, R, S. יש לסמן ירושה באמצעות החץ <---

ב. (1) נתונה המחלקה Tester:

```

public class Tester {
    public static void Main (string[] args) {
        P p = new P (2);
        Q q = new Q (1);
        R r = new R (4);
        S s = new S (3);
        P pq = new Q (2);
    }
}

```

```
Q qr = new R (5);
/**
}
}
```

סרטטו את העצמים שנוצרו בפעולה Main .

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

```
C#
Console.WriteLine (pq.Show(1));           //(1)
Console.WriteLine (qr.Show(r));           //(2)
Console.WriteLine (r.Show(qr));           //(3)
Console.WriteLine (((Q)pq).Show(s));       //(4)
Console.WriteLine (s.Show(pq));           //(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