

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 2
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 **Reading** - מדידה של תחנה מטאורולוגית, ולה שתי תכונות:

· station - מספר התחנה, מטיפוס שלם

· temp - הטמפרטורה שנמדדה (במעלות), מטיפוס שלם

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

רשת תחנות מדידה בגליל שומרת את כל המדידות של יום אחד בתור - qRead מטיפוס **Reading**.

הערות:

· כל תחנה מודדת כמה פעמים ביום, ולכן אותה תחנה יכולה להופיע בתור כמה פעמים.

· המדידות של אותה תחנה יכולות להופיע במקומות שונים בתור.

פער הטמפרטורות של תחנה הוא ההפרש בין הטמפרטורה הגבוהה ביותר לטמפרטורה הנמוכה ביותר שנמדדו בה באותו היום. לתחנה שיש לה מדידה אחת בלבד הפער הוא 0.

למשל, בתור שלפניכם (משמאל - ראש התור) פער הטמפרטורות של תחנה 12 הוא $18 - 7 = 11$, של תחנה 30 הוא $9 - (-2) = 11$ ושל תחנה 7 הוא 0.

7	12	30	7	12	30	12	station
14	21	2-	14	25	9	18	temp

א. (1) ממשו את הפעולה שלפניכם:

Java - `public static Queue<Reading> tempGaps (Queue<Reading> qRead)`

#C - `public static Queue<Reading> TempGaps (Queue<Reading> qRead)`

הפעולה מקבלת תור מדידות - qRead, ומחזירה תור **חדש** של עצמים מטיפוס **Reading**, ובו כל

תחנה מופיעה פעם אחת בלבד: בתכונה station - מספר התחנה, ובתכונה temp - **פער**

הטמפרטורות של התחנה.

בעבור התור שבדוגמה, הפעולה תחזיר תור ובו שלושה עצמים: (7, 12), (11, 30), (0, 7).

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) תחנה "**לא יציבה**" היא תחנה שפער הטמפרטורות שלה **גדול** מן הממוצע של פערי

הטמפרטורות של כל התחנות (כל תחנה נספרת בממוצע פעם אחת).

ממשו את הפעולה שלפניכם:

Java - `public static Queue<Integer> unstable (Queue<Reading> qRead)`

#C - `public static Queue<int> Unstable (Queue<Reading> qRead)`

הפעולה מקבלת תור מדידות של יום אחד - `qRead` (שבו אותה תחנה יכולה להופיע כמה פעמים), ומחזירה תור מטיפוס **שלם** ובו מספרי כל התחנות ה"לא יציבות", כל תחנה פעם אחת (ללא חשיבות לסדר). הניחו שבתור `qRead` יש לפחות מדידה אחת. בעבור התור שבדוגמה: הפערים הם 7, 11, 0, הממוצע הוא 6, ולכן הפעולה תחזיר תור ובו 12 ו-30.

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 שהתקבל.

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

2. (25 נקודות)

"מסגרת" היא רצף של מספרים בתוך סדרה של מספרים, המתחיל במספר כלשהו ומסתיים **באותו** מספר. **משקל המסגרת** הוא סכום המספרים שנמצאים **בין** שני הקצוות שלה (בלי הקצוות). אם שני הקצוות צמודים זה לזה, משקל המסגרת הוא 0.

לדוגמה: בסדרת המספרים 4, 8, 2, 6, 8, 5, 4, 2, 8, 6, 5, 4, 3 יש שתי מסגרות:

i. 4, 8, 2, 6, 8, 5, 4 - משקלה $2 + 8 + 6 + 5 = 29$

ii. 8, 2, 6, 8 - משקלה $2 + 6 = 8$

נתונה שרשרת חוליות `lst` מטיפוס שלם, המכילה מספרים חיוביים. כל מספר מופיע בשרשרת **פעם אחת או פעמיים** (לא יותר).

א. כתבו פעולה ששמה `frameWeight` בשפת Java או `FrameWeight` בשפת #C המקבלת את השרשרת `lst` ומספר שלם `num` **המופיע** בשרשרת.

הפעולה תחזיר את **משקל** המסגרת שהמספר `num` מתחיל ומסיים.

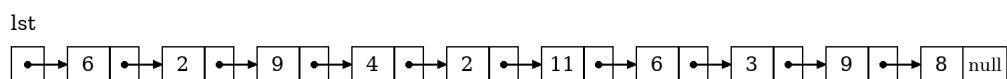
אם `num` מופיע בשרשרת פעם אחת בלבד, הפעולה תחזיר -1.

הערות:

- חובה לשמור על השרשרת `lst`.

- **אין** להשתמש בסעיף זה במערך. פתרון הכולל שימוש במערך לא יזוכה בנקודות.

דוגמאות: עבור השרשרת `lst` שלפניכם:



עבור `num = 6` הפעולה תחזיר 28 (המסגרת: 6, 11, 2, 4, 9, 2, 6, ומשקלה $11 + 2 + 4 + 9 + 2 = 28$).

עבור `num = 2` הפעולה תחזיר 13 (המסגרת: 2, 4, 9, 2).

עבור `num = 11` הפעולה תחזיר -1 (משום ש-11 מופיע בשרשרת פעם אחת בלבד).

ב. כתבו פעולה ששמה `heaviest` בשפת `Java` או `Heaviest` בשפת `C#` המקבלת את השרשרת `lst`. הפעולה תחזיר את המספר שמתחיל ומסיים את המסגרת **הכבדה ביותר** (המסגרת שמשקלה הגדול ביותר). אם יש כמה מסגרות שמשקלן הגדול ביותר, הפעולה תחזיר מביניהן את המספר שההופעה הראשונה שלו קודמת בשרשרת. אם אין בשרשרת שום מסגרת, הפעולה תחזיר `-1`. אפשר להשתמש בפעולה שכתבתם בסעיף א.

הערות:

- בפעולה זו אין חובה לשמור על השרשרת `lst`.
- **אין** להשתמש בסעיף זה במערך. פתרון הכולל שימוש במערך לא יזוכה בנקודות.
- לדוגמה:** עבור השרשרת `lst` שבדוגמה לעיל הפעולה תחזיר `6`.
- הסבר:** בשרשרת שלוש מסגרות: של `6` (משקל `28`), של `2` (משקל `13`) ושל `9` (משקל `26`). המסגרת של `6` היא הכבדה ביותר.

3. (25 נקודות)

שימו לב: בשאלה זו אפשר להשתמש בפעולות של המחלקות `Tower`, `Edge` ו-`Node`, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

"חזקה של 2" היא מספר מהצורה 2^k , כאשר $k \geq 0$ שלם, כלומר אחד מהמספרים `1, 2, 4, 8, 16, 32, ...` (גם המספר `1` הוא חזקה של `2`). לפניהם הפעולה החיצונית `isPowerOfTwo/IsPowerOfTwo`. אפשר להשתמש בפעולה בלי לממש אותה.

כותרת הפעולה	תיאור הפעולה
<code>public static boolean isPowerOfTwo (int num) - Java</code>	הפעולה מחזירה <code>true</code> אם הערך <code>num</code> שהתקבל (<code>num > 0</code>) הוא חזקה של <code>2</code> .
<code>public static bool IsPowerOfTwo (int num) - #C</code>	אחרת היא מחזירה <code>false</code> .

א. ממשו את הפעולה החיצונית שלפניהם:

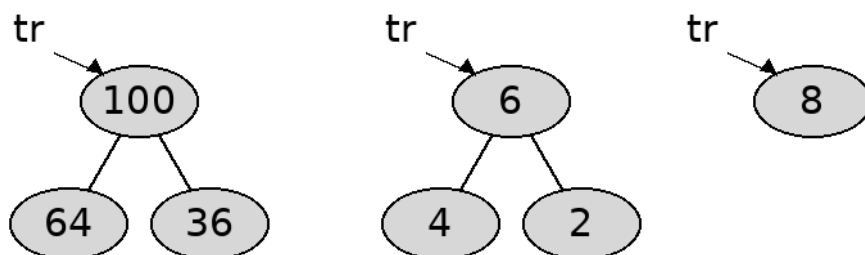
`public static boolean splitPower (BinNode<Integer> tr) - Java`

`public static bool SplitPower (BinNode<int> tr) - #C`

הפעולה מקבלת צומת ללא בנים (עלה) שערכו גדול מ-`0`. אם ערך הצומת הוא חזקה של `2`, הפעולה מחזירה `false`.

אחרת, הפעולה מוסיפה לצומת שני בנים: ערך הבן השמאלי הוא החזקה של `2` הגדולה ביותר שקטנה מערך הצומת, וערך הבן הימני הוא ההפרש בין ערך הצומת לערך הבן השמאלי. לאחר ההוספה הפעולה מחזירה `true`.

דוגמאות: בתום הפעולה ייראו הצמתים כך (עבור הערכים `100, 6, 8`):



(עבור 100: החזקה הגדולה ביותר של 2 שקטנה מ-100 היא 64, ו- $100 - 64 = 36$. הערך 8 הוא חזקה של 2, ולכן לא נוספו לו בנים והפעולה החזירה false.)

ב. (1) נתונה הפעולה build/Build שלפניהם, המקבלת צומת ללא בנים שערכו גדול מ-0. בשפת :

Java

```
public static void build(BinNode<Integer> tr) {
    if (splitPower(tr)) {
        build(tr.getLeft());
        build(tr.getRight());
    }
}
```

C#

```
public static void Build(BinNode<int> tr)
{
    if (SplitPower(tr))
    {
        Build(tr.GetLeft());
        Build(tr.GetRight());
    }
}
```

(1) סרטטו את העץ כפי שהוא יראה בתום הפעולה build/Build עבור צומת ללא בנים - tr שערכו 92. יש להראות מעקב.

(2) הסבירו מה מבצעת הפעולה build/Build.

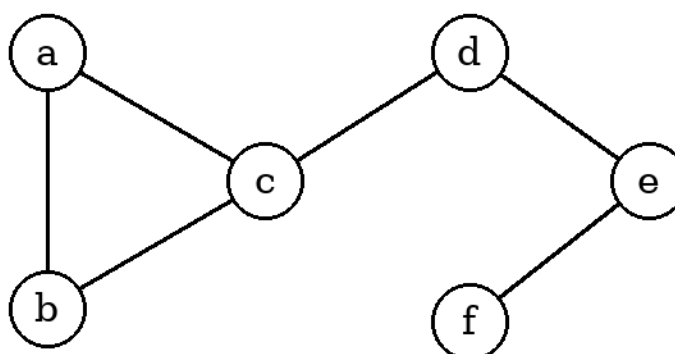
פרק שני (50 נקודות) - אלגוריתמים

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

4. (25 נקודות)

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

א. להלן גרף $G = (V, E)$ שאינו מכוון:



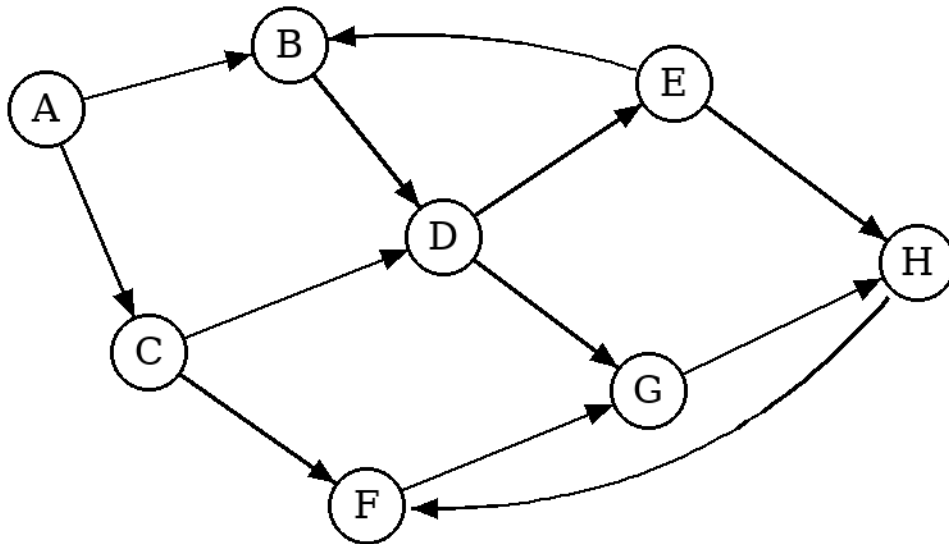
א. (1) האם הגרף קשיר? נמקו.

(2) האם הגרף דו-צדדי? נמקו.

(3) האם הגרף מלא (שלם)? נמקו.

(4) האם הגרף הוא עץ? נמקו.

- ב. (1) במכללה מציעים שמונה קורסים: A, B, C, D, E, F, G, H. לחלק מן הקורסים יש **דרישות קדם**. בסרטוט שלהלן, חץ מקורס X לקורס Y פירושו שקורס X הוא דרישת קדם לקורס Y (כלומר יש לסיים את X לפני שמתחילים את Y):



(1) סרטטו את רשימת הסמיכויות המייצגת את דרישות הקדם בין הקורסים.

(2) בעקבות עדכון של תכנית הלימודים הוחלט לבטל כמה דרישות קדם:

• קורס E - ביטול הדרישה ש-E יילמד לפני B

• קורס H - ביטול הדרישה ש-H יילמד לפני F

• קורס A - ביטול הדרישה ש-A יילמד לפני C

(2) סרטטו את רשימת הסמיכויות המתקבלת לאחר הביטולים המצוינים לעיל.

(3) I. נועה רוצה ללמוד את כל שמונת הקורסים, קורס אחד בכל פעם. לשם כך היא רוצה לבדוק אם אפשר לסדר את הקורסים כך שכל קורס יילמד רק אחרי כל דרישות הקדם שלו, ואם אפשר - למצוא סדר כזה.

מהו האלגוריתם שעליה להשתמש בו למטרות אלה? הסבירו.

(3) II. הפעילו את האלגוריתם שכתבתם עליו בסעיף I(3) על הגרף שהתקבל לאחר הביטולים

(סעיף (2)), וכתבו סדר לימוד של הקורסים שמצאתם.

יש לבצע מעקב מפורט בכל שלב בהתאם לאלגוריתם שהפעלתם.

5. (25 נקודות)

לפניכם שש טענות א-ו. בחרו **בחמש** מהן, וציינו בנוגע לכל טענה שבחרתם אם היא נכונה או לא נכונה. אם הטענה נכונה - נמקו מדוע, ואם הטענה לא נכונה - הביאו דוגמה נגדית.

- א. בגרף G לא מכוון וקשיר הריצו BFS מקודקוד s . לכל קודקוד v , $d[v]$ הוא מספר הקשתות במסלול הקצר ביותר מ- s אל v שה-BFS מצא.
- לכל קשת (u, v) בגרף מתקיים: ההפרש בין $d[u]$ ובין $d[v]$ הוא לכל היותר 1.
- ב. בגרף G ממושקל, לא מכוון וקשיר, שבו כל משקלי הקשתות שונים זה מזה, הקשת שמשקלה הגדול ביותר אינה שייכת לאף עץ פורש מינימלי של הגרף.
- ג. בכל גרף מכוון חסר מעגלים (גמ"ל) יש לפחות קודקוד אחד שאין אף קשת הנכנסת אליו (דרגת הכניסה שלו היא 0).
- ד. נתון גרף G ממושקל ולא מכוון שבו כל המשקלים חיוביים, ונתון בו המסלול הקל ביותר מקודקוד a אל קודקוד b .
- אם נוסיף אותו מספר קבוע $c > 0$ למשקל של כל אחת מהקשתות בגרף, אותו מסלול יישאר בהכרח המסלול הקל ביותר מ- a אל b .
- ה. אם בגרף מכוון חסר מעגלים יש מסלול שעובר בכל הקודקודים של הגרף, אז לגרף יש מיון טופולוגי יחיד.
- ו. בגרף ממושקל, לא מכוון וקשיר, שבו כל המשקלים חיוביים, עץ המסלולים הקצרים ביותר מקודקוד s (העץ המתקבל מהאלגוריתם של דייקסטרה) הוא תמיד גם עץ פורש מינימלי של הגרף.

6. (25 נקודות)

גרף מכוון נקרא "גרף-כמעט-חסר-מעגלים" אם יש בו לפחות מעגל (מכוון) אחד, וקיימת בו קשת שכאשר מורידים אותה מקבלים גרף ללא מעגלים.

בשאלה זו, בכל הגרפים אין קשת מקודקוד לעצמו ואין קשתות כפולות.

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

ב. (1) נתון "גרף-כמעט-חסר-מעגלים" 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 (כשמתעלמים מכיווני הקשתות)? נמקו את תשובתכם. אם עניתם שאפשר, ציינו את מספר רכיבי הקשירות.

ג. (1) נתון גרף מכוון G ובו n קודקודים, המיוצג על ידי רשימת סמיכויות (לכל קודקוד - רשימת הקודקודים שיש אליהם קשת ממנו).

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

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

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 קשר ביניהם. ענו על שני הסעיפים.

א. (1) לפניכם שלוש שפות מעל הא"ב $\{a, b\}$:

$$L_1 = \{ a^n b^m \mid n, m > 0, m = 2n \}$$

$$L_2 = \{ a^n b^m \mid n, m > 0, (n + m) \% 3 = 0 \}$$

$$L_3 = \{ a^n b^m \mid n, m > 0, m < 20 \}$$

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

(1) בנוגע **לכל אחת** מן השפות L_1 - L_3 , 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 $L_1 \cap \bar{L}_3$ רגולרית? נמקו את תשובתכם.

(4) האם השפה $L_1 \cup L_2$ רגולרית? נמקו את תשובתכם.

ב. (1) נתונה השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{ a^m b^k c^x \mid m, k, x > 0 \text{ וגם } m = 1 \text{ או } (k + x) \% 2 = 0 \}$$

הסבר: השפה L מכילה מילים שבהן כל אחת מן האותיות abc מופיעה פעם אחת לפחות, ולפי סדר זה (המופעים של a תחילה, אחר כך של b ואחר כך של c). כמו כן, האות a מופיעה פעם אחת בלבד, **או** שמספר המופעים של b ושל c יחד הוא זוגי.

דוגמה למילה בשפה L היא $abbbcc$, כי כל אחת מן האותיות abc מופיעה בה, ולפי הסדר הנדרש, והאות a מופיעה בה פעם אחת.

דוגמה נוספת למילה בשפה L היא $aabbcc$, כי כל אחת מן האותיות abc מופיעה בה, ולפי הסדר הנדרש, ומספר המופעים של b ושל c יחד הוא 4 - זוגי.

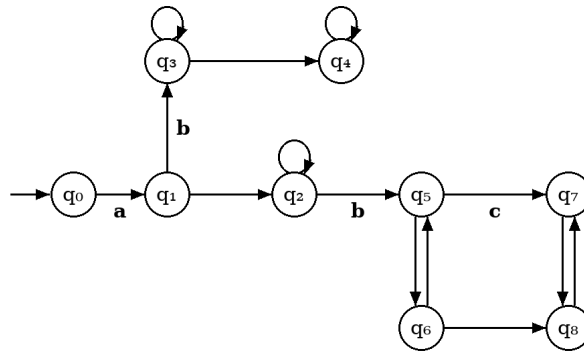
דוגמה למילה שאינה בשפה L היא $aabcc$. אומנם כל אחת מן האותיות abc מופיעה בה, ולפי הסדר הנדרש, אך האות a מופיעה בה פעמיים, ומספר המופעים של b ושל c יחד הוא 3 - אי-זוגי.

(1) לפניכם חמש מילים: $aaabc$, $abcbc$, $aabbbcc$, $abbbbbc$, $aacc$.

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. מצב ההתחלה הוא q_0 . 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 תזכה בנקודות.**



8. (25 נקודות)

L היא שפת כל המילים מעל הא"ב $\{0, 1\}$ המקיימות את כל הדרישות שלפניכם:

– המילה מתחילה באות 1.

– האות השלישית מסוף המילה היא 0 (למשל, במילה 1011 האות השלישית מהסוף היא 0).

– הרצף 11 מופיע במילה **בדיוק פעם אחת**. מופעים חופפים נספרים כל אחד בנפרד; למשל, במילה 0111 הרצף 11 מופיע פעמיים.

– מספר המופעים של האות 0 במילה מתחלק ב-3 בלי שארית.

א. (1) הביאו דוגמה אחת למילה השייכת לשפה L , שאורכה 7 אותיות לפחות.

(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.

9. (25 נקודות)

כתבו מכונת טיורינג המקבלת בתחילת הסרט מספר אונרי הגדול מאפס, ומחזירה מספר חדש כמפורט לפניכם:

- אם המספר שהתקבל מתחלק ב-3 ללא שארית - המכונה מחזירה את המספר המקורי חלקי 3.
- אם המספר שהתקבל אינו מתחלק ב-3 - המכונה מחזירה את השארית של חלוקת המספר המקורי ב-3.

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 המתחלק ב-3

הסרט לפני ההרצה:

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

הסרט לאחר ההרצה:

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

דוגמה למספר שאינו מתחלק ב-3

הסרט לפני ההרצה:

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

הסרט לאחר ההרצה:

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

כתבו מכונת טיורינג המקבלת בתחילת הסרט מספר אונרי הגדול מאפס, ומחזירה מספר חדש כמפורט לעיל.

(25 נקודות)

תכנות מונחה עצמים

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

10. (25 נקודות)

נוסח Java

בנמל "חוף הצפון" פותחה מערכת ממוחשבת לניהול המשאיות הממתינות בשער הנמל, ובה המחלקות האלה:

Container - מכולה, **Reefer** - מכולת קירור, **Tank** - מכולת נוזלים, **Flat** - מכולה פתוחה, **Truck** - משאית, **Gate** - שער הנמל.

להלן פירוט תכונות המחלקות:

למחלקה **Container** (מכולה) שתי תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבעלותה המכולה (מטיפוס מחרוזת).

למחלקה **Reefer** (מכולת קירור) שלוש תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבעלותה המכולה (מטיפוס מחרוזת), **temp** - טמפרטורת הקירור במעלות (מטיפוס שלם).

למחלקה **Tank** (מכולת נוזלים) שלוש תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבעלותה המכולה (מטיפוס מחרוזת), **liquid** - סוג הנוזל שבמכולה (מטיפוס מחרוזת).

למחלקה **Flat** (מכולה פתוחה) שלוש תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבעלותה המכולה (מטיפוס מחרוזת), **covered** - האם המטען מכוסה ביריעה (מטיפוס בוליאני. אם המטען מכוסה - **true**, ואם לא - **false**).

למחלקה **Truck** (משאית) שתי תכונות: **driver** - שם הנהג (מטיפוס מחרוזת), **cont** - המכולה שהמשאית מובילה (מטיפוס **Container**). כל משאית מובילה מכולה אחת בלבד.

למחלקה **Gate** (שער הנמל) שתי תכונות: **arrT** - מערך של המשאיות הממתינות בשער (מטיפוס **Truck**), **count** - כמות המשאיות השמורות במערך (מטיפוס שלם).

המשאיות מופיעות ברצף מתחילת המערך ללא **null** ביניהן, אך ייתכן שהמערך אינו מלא עד הסוף. **א. (1)** סרטוט 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. 1. כתבו במחלקה **Gate** פעולה ששמה **coldestDriver**, המחזירה את שם הנהג של המשאית המובילה את מכולת הקירור שהטמפרטורה שלה היא הנמוכה ביותר. אם אין בשער משאית המובילה מכולת קירור, הפעולה תחזיר **null**.

הניחו שאין בשער שתי מכולות קירור עם אותה טמפרטורה.

ג. 1. כתבו במחלקה **Gate** פעולה ששמה **arrange**, המסדרת מחדש את המשאיות במערך **arrT** כך שיופיעו קודם המשאיות המובילות מכולות קירור, אחר כך המשאיות המובילות מכולות נוזלים, ואחר כך המשאיות המובילות מכולות פתוחות.

הניחו שכל משאית בשער מובילה מכולת קירור, מכולת נוזלים או מכולה פתוחה.

בסוף הפעולה המשאיות יופיעו ברצף בתחילת המערך (ללא **null** ביניהן), וכמותן לא תשתנה.

הפעולה תחזיר את מספר המשאיות המובילות מכולות קירור.

הערה: אין חשיבות לסדר המשאיות בתוך כל סוג. מותר להשתמש במערך עזר.

נוסח C#

בנמל "חוף הצפון" פותחה מערכת ממוחשבת לניהול המשאיות הממתינות בשער הנמל, ובה המחלקות האלה:

Container - מכולה, **Reefer** - מכולת קירור, **Tank** - מכולת נוזלים, **Flat** - מכולה פתוחה, **Truck** - משאית, **Gate** - שער הנמל.

להלן פירוט תכונות המחלקות:

למחלקה **Container** (מכולה) שתי תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבבעלותה המכולה (מטיפוס מחרוזת).

למחלקה **Reefer** (מכולת קירור) שלוש תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבבעלותה המכולה (מטיפוס מחרוזת), **temp** - טמפרטורת הקירור במעלות (מטיפוס שלם).

למחלקה **Tank** (מכולת נוזלים) שלוש תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבבעלותה המכולה (מטיפוס מחרוזת), **liquid** - סוג הנוזל שבמכולה (מטיפוס מחרוזת).

למחלקה **Flat** (מכולה פתוחה) שלוש תכונות: **code** - קוד המכולה (מטיפוס מחרוזת), **owner** - שם החברה שבבעלותה המכולה (מטיפוס מחרוזת), **covered** - האם המטען מכוסה ביריעה (מטיפוס בוליאני. אם המטען מכוסה - **true**, ואם לא - **false**).

למחלקה **Truck** (משאית) שתי תכונות: **driver** - שם הנהג (מטיפוס מחרוזת), **cont** - המכולה שהמשאית מובילה (מטיפוס **Container**). כל משאית מובילה מכולה אחת בלבד.

למחלקה **Gate** (שער הנמל) שתי תכונות: **arrT** - מערך של המשאיות הממתינות בשער (מטיפוס **Truck**), **count** - כמות המשאיות השמורות במערך (מטיפוס שלם).

המשאיות מופיעות ברצף מתחילת המערך ללא **null** ביניהן, אך ייתכן שהמערך אינו מלא עד הסוף. **א. (1)** סרטוט 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 **Gate** פעולה ששמה **ColdestDriver**, המחזירה את שם הנהג של המשאית המובילה את מכולת הקירור שהטמפרטורה שלה היא הנמוכה ביותר. אם אין בשער משאית המובילה מכולת קירור, הפעולה תחזיר **null**.

הניחו שאין בשער שתי מכולות קירור עם אותה טמפרטורה.

ג. כתבו במחלקה **Gate** פעולה ששמה **Arrange**, המסדרת מחדש את המשאיות במערך **arrT** כך שיופיעו קודם המשאיות המובילות מכולות קירור, אחר כך המשאיות המובילות מכולות נוזלים, ואחר כך המשאיות המובילות מכולות פתוחות.

הניחו שכל משאית בשער מובילה מכולת קירור, מכולת נוזלים או מכולה פתוחה.

בסוף הפעולה המשאיות יופיעו ברצף בתחילת המערך (ללא **null** ביניהן), וכמותן לא תשתנה.

הפעולה תחזיר את מספר המשאיות המובילות מכולות קירור.

הערה: אין חשיבות לסדר המשאיות בתוך כל סוג. מותר להשתמש במערך עזר.

נוסח Java

לפניכם שלושה סעיפים א-ג שאין קשר ביניהם. **ענו על שלושת הסעיפים.**

א. לפניכם המחלקה **Card** (קלף) ללא מימוש הפעולות הבונות, והמחלקה **Joker** (ג'וקר) ללא מימוש הפעולות הבונות ופעולת **toString**.

Java

```
public class Card {
    private static int total = 100;
    protected int score;
    public Card () {...}
    public Card (int score) {...}
    public String toString() {return "total=" + total + ", score=" + score;}
}

-----

public class Joker extends Card {
    private boolean wild;
    public Joker (boolean wild) {...}
    public Joker (int score, boolean wild) {...}
    public String toString() {...}
}
```

לפניכם קטע קוד והפלט שהתקבל בעקבות הרצת קטע הקוד:

הקוד

Java

```
Card c1 = new Card (7);
System.out.println (c1);           //1
Card c2 = new Card ();
System.out.println (c2);           //2
Card c3 = new Joker (true);
System.out.println (c3);           //3
Card c4 = new Joker (3, false);
System.out.println (c4);           //4
```

הפלט

```
total=107, score=7                //1
total=108, score=1                //2
total=108, score=0, wild=true     //3
total=111, score=3, wild=false    //4
```

ממשו את הפעולות הבונות במחלקה **Card** ובמחלקה **Joker**, ואת הפעולה **toString** במחלקה **Joker**, בהתאם לעקרונות תכנות מונחה עצמים, כך שיתקבל הפלט הנתון. אין להוסיף פעולות נוספות למחלקות. תשובה הכוללת פעולות נוספות במחלקות, לא תזכה בנקודות.

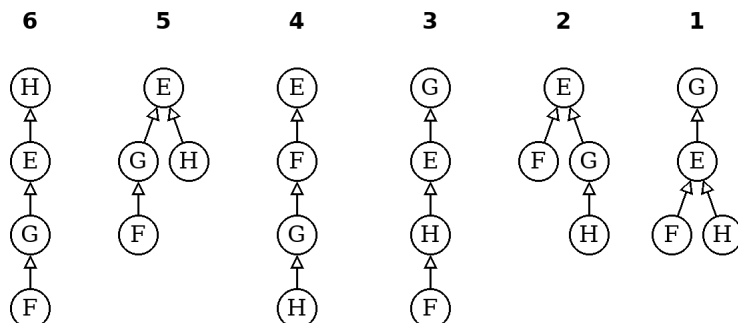
ב. (1) נתונות המחלקות **E, F, G, H** והפעולה **main** במחלקה **Tester**, **שרצה באופן תקין**.

Java

```
public class Tester {
    public static void main(String[] args) {
        E e1 = new H();
        G g1 = new F();
        E e2 = new F();
    }
}
```

```
}
}
```

לפניכם שישה תרשימי הייררכייה 1-6 (החץ מסמן יחס ירשה). בנוגע לכל אחד מהם, ציינו אם ייתכן או לא ייתכן שהוא מתאר את יחס ההייררכייה בין המחלקות E, F, G, H. אם ציינתם שלא ייתכן, נמקו מדוע.



(2) בסוף הפעולה main נוספו שורות הקוד שלהלן, שרצות באופן תקין:

Java

```
E e3 = new F();
H h1 = (H)e3;
```

בעקבות זאת, האם תשתנה תשובתכם בתת-סעיף ב(1)? נמקו את תשובתכם.

ג. נתונה המחלקה **OnlineCourse** (קורס מקוון), היורשת מהמחלקה **Course** (קורס):

Java

```
public class OnlineCourse extends Course {
    private String link;
    public OnlineCourse (String link) {
        this.link = link;
    }
    public OnlineCourse (String title, int hours, String link) {
        super (title, hours);
        this.link = link;
    }
    public int price () {return super.price() - Course.discount; }
    public boolean isLong () {return this.hours > 30; }
    public String describe (int n) {return this.describe() + this.link.substring(0,
n); }
}
```

כתבו במחלקה **Course** את **תכונות** המחלקה (כולל הרשאות הגישה) ואת **הכותרות** של הפעולות הנדרשות כדי שהמחלקה **OnlineCourse** תהיה תקינה, בהתאם לעקרונות תכנות מונחה עצמים. אין צורך לממש את הפעולות שאת הכותרות שלהן הוספתם במחלקה **Course**. אין לשנות את הקוד של המחלקה **OnlineCourse**.

נוסח C#

לפניכם שלושה סעיפים א-ג שאין קשר ביניהם. ענו על שלושת הסעיפים.

א. לפניכם המחלקה **Card** (קלף) ללא מימוש הפעולות הבונות, והמחלקה **Joker** (ג'וקר) ללא מימוש הפעולות הבונות ופעולת **ToString**.

C#

```
public class Card {
    private static int total = 100;
    protected int score;
    public Card () {...}
    public Card (int score) {...}
    public override string ToString() {return "total=" + total + ", score=" +
score;}
}
-----
public class Joker : Card {
    private bool wild;
    public Joker (bool wild) {...}
    public Joker (int score, bool wild) {...}
    public override string ToString() {...}
}
```

לפניכם קטע קוד והפלט שהתקבל בעקבות הרצת קטע הקוד:

הקוד

```
C#
Card c1 = new Card (7);
Console.WriteLine (c1);           //1
Card c2 = new Card ();
Console.WriteLine (c2);           //2
Card c3 = new Joker (true);
Console.WriteLine (c3);           //3
Card c4 = new Joker (3, false);
Console.WriteLine (c4);           //4
```

הפלט

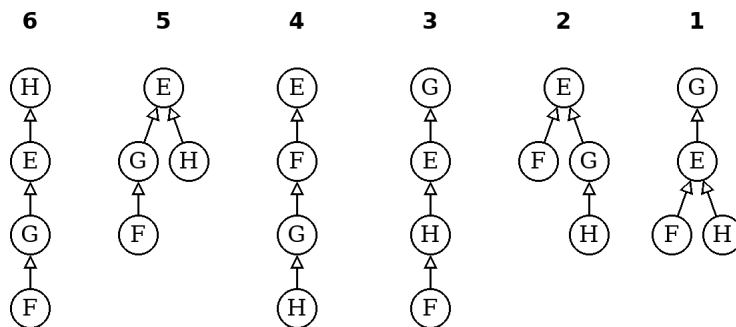
```
total=107, score=7                //1
total=108, score=1                //2
total=108, score=0, wild=True     //3
total=111, score=3, wild=False    //4
```

ממשו את הפעולות הבונות במחלקה Card ובמחלקה Joker, ואת הפעולה ToString במחלקה Joker, בהתאם לעקרונות תכנות מונחה עצמים, כך שיתקבל הפלט הנתון. אין להוסיף פעולות נוספות למחלקות. תשובה הכוללת פעולות נוספות במחלקות, לא תזוכה בנקודות.

ב. (1) נתונות המחלקות E, F, G, H והפעולה Main במחלקה Tester, **שרצה באופן תקין**.

```
C#
public class Tester {
    public static void Main (string[] args) {
        E e1 = new H();
        G g1 = new F();
        E e2 = new F();
    }
}
```

לפניכם שישה תרשימי הייררכייה 1-6 (החץ מסמן יחס ירושה). בנוגע לכל אחד מהם, ציינו אם ייתכן או לא ייתכן שהוא מתאר את יחס ההייררכייה בין המחלקות E, F, G, H. אם ציינתם שלא ייתכן, נמקו מדוע.



(2) בסוף הפעולה Main נוספו שורות הקוד שלהלן, **שרצות באופן תקין**:

```
C#
E e3 = new F();
H h1 = (H)e3;
```

בעקבות זאת, האם תשתנה תשובתכם בתת-סעיף ב(1)? נמקו את תשובתכם.

ג. נתונה המחלקה **OnlineCourse** (קורס מקוון), היורשת מהמחלקה **Course** (קורס):

```
C#
public class OnlineCourse : Course {
    private string link;
    public OnlineCourse (string link) {
        this.link = link;
    }
    public OnlineCourse (string title, int hours, string link) : base (title, hours)
    {
        this.link = link;
    }
    public override int Price () {return base.Price() - Course.discount;}
    public bool IsLong () {return this.hours > 30;}
    public string Describe (int n) {return this.Describe() + this.link.Substring(0,
n);}
}
```

כתבו במחלקה **Course** את **תכונות** המחלקה (כולל הרשאות הגישה) ואת **הכותרות** של הפעולות הנדרשות כדי שהמחלקה **OnlineCourse** תהיה תקינה, בהתאם לעקרונות תכנות מונחה עצמים. **אין צורך לממש את הפעולות שאת הכותרות שלהן הוספתם במחלקה Course. אין לשנות את הקוד של המחלקה OnlineCourse.**

12. (25 נקודות)

נוסח Java

נתונות המחלקות Opal , Gem :

Java

```
public class Gem {
    private int w;

    public Gem() { this(1); }
    public Gem(int w) { this.w = w; }
    public int getW() { return this.w; }
    public void shine() { this.w = this.w + 1; }
    public void polish() {
        shine();
        this.w = this.w * 2;
    }
    public void join(Gem g) { this.w = this.w + g.getW(); }
    public String toString() { return "Gem " + this.w; }
}

public class Opal extends Gem {
    private int c;

    public Opal(int w, int c) {
        super(w);
        this.c = c;
    }
    public Opal(int c) {
        this(c * 2, c);
    }
    public void shine() { this.c = this.c + getW(); }
    public void join(Opal o) { this.c = this.c + o.c; }
    public String toString() { return "Opal " + getW() + "/" + this.c; }
}
```

א. לפניכם קטע קוד:

Java

```
Gem[] g = new Gem[5];
g[0] = new Gem();
g[1] = new Opal(3);
g[2] = new Opal(4, 1);
g[3] = new Gem(6);
g[4] = g[2];
for (int i = 0; i < g.length; i++)
    System.out.println(g[i]);
```

ציירו את העצמים שנוצרו, וכתבו מה הקוד מדפיס.

ב. לפניכם המשך קטע הקוד:

Java

```
g[0].polish();
g[1].polish();
g[2].join(g[1]);
Opal p = (Opal) g[1];
p.join((Opal) g[4]);
p.join(g[3]);
((Opal) g[4]).join(p);
g[3].polish();
```

```
System.out.println("p: " + p);
for (int i = 0; i < g.length; i++)
    System.out.println(g[i]);
```

כתבו מה הקוד מדפיס.

נוסח #C

נתונות המחלקות Opal , Gem :

```
C#
public class Gem {
    private int w;

    public Gem() : this(1) { }
    public Gem(int w) { this.w = w; }
    public int GetW() { return this.w; }
    public virtual void Shine() { this.w = this.w + 1; }
    public void Polish() {
        Shine();
        this.w = this.w * 2;
    }
    public void Join(Gem g) { this.w = this.w + g.GetW(); }
    public override string ToString() { return "Gem " + this.w; }
}

public class Opal : Gem {
    private int c;

    public Opal(int w, int c) : base(w) {
        this.c = c;
    }
    public Opal(int c) : this(c * 2, c) { }
    public override void Shine() { this.c = this.c + GetW(); }
    public void Join(Opal o) { this.c = this.c + o.c; }
    public override string ToString() { return "Opal " + GetW() + "/" + this.c; }
}
```

א. לפניכם קטע קוד:

```
C#
Gem[] g = new Gem[5];
g[0] = new Gem();
g[1] = new Opal(3);
g[2] = new Opal(4, 1);
g[3] = new Gem(6);
g[4] = g[2];
for (int i = 0; i < g.Length; i++)
    Console.WriteLine(g[i]);
```

ציירו את העצמים שנוצרו, וכתבו מה הקוד מדפיס.

ב. לפניכם המשך קטע הקוד:

```
C#
g[0].Polish();
g[1].Polish();
g[2].Join(g[1]);
Opal p = (Opal) g[1];
```

```
p.Join((Opal) g[4]);  
p.Join(g[3]);  
((Opal) g[4]).Join(p);  
g[3].Polish();  
Console.WriteLine("p: " + p);  
for (int i = 0; i < g.Length; i++)  
    Console.WriteLine(g[i]);
```

כתבו מה הקוד מדפיס.

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